8:16

The Alternative Atari Newsletter

ATARI

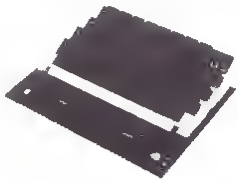
BAPAUG

£1.25 | 3rd Quarter 1991 | Issue 13

Supercharger Review

Home Filing Manager Temper Saver
MIDI In Practice
British SpartaDOS Time
Fun with Stereo Sound on the TT030 and STE
34th Longleat Radio Rally

Choosing a memory upgrade for your Atari ST just got easier!



Once Frontier's XTRA-RAM *Deluxe* is installed in your ST you can upgrade in stages to 4MB whenever you like - it couldn't be easier. Includes a two year guarantee and a no-quibble ten day money back offer

Is your ST always running out of memory? Frontier Software's new XTRA RAM Deluxe is the easy to fit and simple to upgrade memory expansion that you've been waiting for

Thousands of ST users have already upgraded their STs to 1MB or 2½MB using the original XTRA RAM in their homes and offices without having to send their computers away. Now Frontier, makers of the original XTRA RAM, introduce their new memory upgrade - The XTRA-RAM Deluxe.

The experience, quality and expertise that went into the best selling XTRA-RAM has been applied to the new XTRA-RAM Deluxe. The XTRA RAM Deluxe will upgrade your Atari STFM to 1MB then 2½MB and then 4MB with easy to install SIMM memory cards. Your Mega 1ST can be upgraded to 2MB and then to 4MB. Mega 2STs can be expanded to the full 4MB. Once the XTRA-RAM Deluxe is installed in your ST you can choose when to upgrade further - you simply change or add SIMM boards - just like the ST^E. You can even use these SIMM boards if you later upgrade to the ST^E.

Easy To Fit

Installation of the XTRA-RAM Deluxe couldn't be easier. Following the instructions laid out in our simple to follow manual which is written with the non-technical reader in mind - you disassemble your STFM or Mega ST, plug the XTRA-RAM Deluxe into two places inside your computer and reassemble it. The whole process takes around 1½ hours and most STs will not require any soldering whatsoever.

The XTRA-RAM Deluxe fits all MMU chips (including the 100109) whether they are soldered down or not. If your ST has a soldered down Video Shifter chip or a 101601 type MMU then some soldering will be required. You can do this soldering yourself or Frontier or your dealer can do it for you for a small extra charge. If your ST has a socketed Video Shifter chip and an MMU which isn't the 101601 type, then you can fit the XTRA-RAM Deluxe without any soldering whatsoever. If you need any further explanation of this, contact Frontier direct.

No Need To Stop At 1MB or 2½MB

Most memory upgrades for the Atari STs will give you a simple upgrade to 1MB for about the same price as the XTRA-RAM Deluxe. Some of these upgrades may be as easy to fit as the XTRA RAM Deluxe, but they do not allow you to expand your ST further. Once the XTRA RAM Deluxe is installed in your ST, all you have to do to upgrade further to 2½MB or 4MB is install extra SIMM memory cards into the empty sockets on the XTRA RAM Deluxe. This is a very simple process which takes less than ½ hour.

Don't restrict yourself to just 1MB. Make sure that you choose an upgrade like the XTRA-RAM Deluxe, which can grow with your needs.

Totally Compatible

The extra memory that the XTRA RAM Deluxe gives your ST is totally compatible with all of your ST programs. The ST's memory controller chip logs in the extra memory and makes it available for your programs. You will automatically get extra memory for DTP, word processing, MIDI, running the Atari Laser printer and everything else that you use your ST for.

Software Included

Each XTRA-RAM Deluxe upgrade is supplied with free RAM disk and printer spooler software and a free RAM testing program so that you know with certainty that your installation has worked.

Satisfaction Guaranteed

The XTRA-RAM Deluxe is supplied under Frontier's ten day money back offer - if you aren't satisfied with the XTRA-RAM Deluxe for any reason you can return it to the place of purchase within ten days for a full refund (participating resellers only). The XTRA-RAM Deluxe also carries a full two years guarantee. The XTRA RAM Deluxe is designed and manufactured in the UK. Unlike some ST upgrades, Frontier guarantee that each XTRA RAM Deluxe uses only brand new SIMM memory boards making your ST and its memory upgrade more reliable both now and in the future.

Prices

Frontier supply the XTRA-RAM Deluxe in four ways:

- XTRA-RAM Deluxe Unpopulated (without memory) £39.99
- XTRA-RAM Deluxe ½MB upgrades your 520STFM to 1MB (can later upgrade your STFM to 2½MB and then 4MB) £69.99
- XTRA-RAM Deluxe 2MB upgrades your 520STFM, 1040ST to 2½MB, Mega 1ST to 2MB or your Mega 2ST to 4MB (can later upgrade your ST or Mega 1ST to 4MB) £119.99
- XTRA-RAM Deluxe 4MB upgrades your 520STFM, 1040ST and Mega 1ST to 4MB £199.99

Please add £3.00 postage and packing to all orders under £120.00. All prices include VAT. Frontier accepts payment by Visa, Access, Switch, cheque or postal order. Prices subject to change without notice. Goods subject to availability. Specification subject to change without notice.

Designed and Manufactured in the UK. Frontier also manufactures memory upgrades for the Atari ST^E

Available from your local dealer or direct from:

2 Year Guarantee

Frontier Software

10 DAY MONEY BACK GUARANTEE

P.O. Box 113, Harrogate, North Yorkshire England HG2 0BE. Phone (0423) 567140/530577. Fax (0423) 522874

We've finally sorted out the editorial team (no more silly names to cover the fact that I did most of the work!). So here it is!

Colin Hunt ..... Editor
248 Wimborne Road, Oakdale, Poole, Dorset BH15 3EF

Paul Brookes ..... ST Editor
32 Duddsway Road, West Parley, Wimborne, Dorset BH22 8RE

Ian Brooker ..... 8 Bit Editor
163 Verity Crescent, Canford Heath, Poole, Dorset BH17 7TX

Please send 8 bit articles to Ian, ST to Paul and all the rest to me (Colin). With this new editorial setup sharing the load, I may now have some time to write articles as well!

8:16 is produced by the 8:16 SIG within the BaPAUG for the enjoyment of everyone who loves their Atari computer.

Printing organised by Apple Art Associates, design and print consultants, 0202 669094

If you can send articles / letters electronically we can also be mailed at the X.400 ID (C:GB ADMD:TMAIL.UK, O:SPRINTINTL, FN:Colin SN:Hunt) If you understand it, you'll probably know how to use it!

This issue of 8:16 was produced using an Atari 130XE 1050 disk drive 520STFM, SM124 monitor Cumana 1M external drive, Apple Macintosh SE Star LC24/10 and Laserwriter Plus. The software used was AtariWriter Plus on the 8 bit, 1st Word Plus, Gem Kermit and ST Xformer II on the ST. Microsoft Word, MacPaint, MacDraw, Mac Kermit, LQ Grabber and Aldus Pagemaker Version 4 on the Macintosh.
Files are transferred from the 8bit to the ST by directly reading the 8bit disks using a 1050 disk drive connected to the ST printer port and saving the files on a ST disk, with help from the Xformer II software. Issue 14 will have an article on how this is done. Standard old Kermit is used to transfer these files to the Mac (after some work has been done on them!)

The BaPAUG is a non profit making organisation.

Copy date for the next issue of 8:16 is 21st October, 1991
Issue date is 9th November, 1991

Contents

Features

Regulars

8 Bit Articles ST Articles

The Bournemouth and Poole Atari User Group is a member of The Association of Atari User Groups.

Notice Board

AMS 5

For those of you at the last AMS show (#4) held last November you will be pleased to here that there is to be a show again this year

For those of you who missed it, the All Micro Show (Alternative Micro Show as it was called last year) is the ONLY show where you can find a large selection of Atari 8 bit products on several different stands.

At this years show there will be stands occupied by Page 6, Dean Garraghty, Grafin International/BaPAUG and Micro Discount.

The show is being held on the 9th November at Bingley Hall, Stafford.

If you are serious about your 8 bit, be there.

New Disk Based Newsletter

Dean Garraghty, who started by writing PD software for the Atari 8 bit, then commercial programs has recently released a disk based newsletter called **NEWS-DISK**. Issues 1 & 2 have so far been released and look very impressive. As well as featuring ready to run software the disk includes an easy to use text file reader and program to print these files in either 40 or 80 columns. A subscription cost £4.00 for 4 issues (1 year?). Dean will also be at the up and coming All Micro Show 5, being held at Bingley Hall, Stafford on 9th November, 1991

Dean Garraghty
62 Thomson Avenue, Balby, Doncaster, DN4 0NU

Competition Winners

I would like to thank everyone who entered the competition within the last issue of 8:16. It went so well that we are planning a new competition for the next issue, which will also contain the solution to this one and a little surprise. For the moment, here are the winners:

Forget-Me-Clock II	Tim Bicknell, Haslemere, Surrey
US Doubler	Mark Horobin, Brownshill, West Midlands
BASIC XL	D. Naylor, Coulsdon, Surrey
Subscriptions.	A. Gerum, Peterlee, County Durham
	D. Swingler, Welling, Kent

Colour Deskjet

Well, not exactly. **System Insight** have released a range of colour cartridges and refill kits for the Deskjet and BJ10e printers. For a 'full colour printing' system you will need four cartridges - Cyan, Magenta, Yellow and Black and special colour separation software. The paper is then passed through the printer four times to produce the final output. The sample we were sent is very good. The cartridges cost £25.00 (other colours are available), while the refill kits are £14.98 each. These prices include VAT and P&P

System Insight
120 Wordsworth Court, Middlefield, Hatfield, Herts AL10 0EP
Telephone: (0707) 276913

WANTED

Do you produce Atari related products?
Want some free advertising?
Then send a press release or information to 8:16 and we will pin it up on the Notice Board

Frontier Software
Lower Prices AGAIN

Frontier Software have once again lowered the retail prices on their ST products. The new prices are as follows:

Printer-Q 128K	£49.99
Printer-Q 256K	£59.99
Printer-Q 512K	£79.99
Printer-Q 1MB	£99.99
Xtra-RAM ST Unpopulated	£29.99
Xtra-RAM ST 512K	£59.99
Xtra-RAM ST 2MB	£99.99
Xtra-RAM ST^E 512K	£39.99
Xtra-RAM ST^E 2MB	£89.99
Xtra-RAM ST^E 4MB	£169.99
Forget-Me-Clock II	£24.99

As well as these new lower prices Frontier have released a new product called the Xtra-RAM Deluxe which is a STF/STFM and Mega ST memory expansion which uses SIMM memory boards. The retail prices for these upgrades are:

Xtra-RAM Deluxe Unpopulated	£39.99
Xtra-RAM Deluxe 0.5MB	£69.99
Xtra-RAM Deluxe 2MB	£119.99
Xtra-RAM Deluxe 4MB	£199.99

Frontier Software
P.O Box 113, Harrogate,
North Yorkshire, HG2 0BE
Telephone (0423) 567140/530577

8:16 Subscription Rates

Annual (4 issues)

U.K.	£4.80
Europe	£9.00
Elsewhere (sea)	£9.00
Elsewhere (air)	£16.00

8:16 Commercial Advertisement Rates

Full page	£45.00
Half page	£25.00
Quarter page	£15.00

Phone (0202) 677895 for details

Dear 8:16,

ST Reads 8-bit Disks!

In 8:16 it says that files can be ported from the 8-bit disks by hooking up to the STs printer port. I now have an 810 drive - is there any chance of finding out how?

B. Capel
GFA User

I transfer the files using Darek Mihocka's Xformer software which emulates the 130XE. For full emulation you need versions 2.3 (which requires 1MB of RAM, and supports VBIs etc) and 2.55 (which runs in 512K machines). Version 2.55 is easier to use, but supports less features. Both versions are capable of reading disks within Atari 8 bit disks drives via a cable that connects the printer port of the ST to the SIO port of the disk drive. The cable, along with both versions of the software are available from Grafin International (see their advert within this issue).

Once you have everything set up, configure the emulator (version 2.55) to use an external drive for D1 and then RESET the emulator with a DOS disk in your 810 (or 1050 etc). At the ready prompt type DOS<RET> and you will be at the DOS menu. If a DOS 2.x compatible you can transfer files from the 8 bit disk using the file copy option (C). The ST disk is device F. Note, that this device handler only supports data I/O, there are no facilities to rename files, delete files etc. In fact, sometimes with DOS 2.5 if you try and get a directory of device F the emulator hangs!

If you wish to transfer BASIC programs, load the BASIC program while at the Ready prompt (LOAD "D:fname.ext") and list to the F device (LIST "F:C:\SOURCE\fname.lst"). Note, that after F: you can use any legal ST filename.

Back Issues

Thank you for sending me my first issue of 8:16.

I was particularly interested to read the article on the Turbo 816 from Dataque - this was the first time I had heard of this unit and it has whetted my curiosity. Is it possible to obtain copies of the previous articles in the series? In fact, I would be interested in purchasing previous issues of 8:16 if they are available - perhaps you could let me know.

Terry Chamberlain
Macclesfield

Sorry, we have no back issues of 8:16. However, we are planning to offer a re-print service. Hopefully you will find an information sheet with this issue. Would you be interested in upgrading your Atari 8 bit to 16 bits? If so, how much would you be willing to pay? Tell Grafin International, as they are thinking of importing this product, but need to know if there is a potential market.

MIDI In Practice

The 8:16 is very good. But please could you try (and this is where you would beat the 'New Atari User') and publish an article on creating music on your Atari like the pro's do.

G. Swales
Harrogate

This letter arrived about 2 weeks after the 'MIDI In Practice' article arrived! We are very keen to publish articles that demonstrate practical uses for computers as this helps to educate users and show that they are more than just toys to play games on.

Packet Radio

I am a radio Amateur and am looking for a suitable 130XE driver program for Packet Radio. I wandered if you may have one in your library.

J. Faviconbridge
Halstead

Sorry, we do not have a suitable program. Can anyone help?

Lynx Game Review

Ninja Gai Den
Atari Corp. £29.99
Reviewed by Thomas Holzer

Here we have the first beat 'em up on the Lynx, "Ninja Gai Den" or Shadow Warriors as it was called in GB, and very good it is too.

Fans of this game will be familiar with the plot, but put simply it is a "run and jump and kill everything that moves" type of game.

You play the part of the ninja running along the screen kicking the hell out of the enemies, sometimes five at a time. By pushing the baddies into telephone boxes!!! or wooden crates you can pick up a bonus like a sword, extra health or even an extra life, all of which you shall certainly be needing.

Animation of your little ninja is very well done indeed, I like the movement how he swings along pipes and then somersaults himself onto them. Also the run and jump animation is pixel perfect.

The enemies need to be hit more than once to dispose of them, so sometimes it can become a bit crowded and you lose sight of where you are, but on the other hand it is possible to kick two or three at a time.

At the end of each level you will encounter a guardian, they're big, strong and very hard to kill, so make sure you have some extra weapons handy.

On completion of a level you are greeted with a picture of the ninja, for example after level 1 you'll see the ninja recovering in hospital reading the news, very good.

The music is different each level, adding to the atmosphere and the fx are noisy and crunchy.

I also want to mention the manual coming with the game, it is a large poster in the style of a japanese comic, it reminds me of the old story line manuals from VCS games, remember "Yars revenge?"

To sum this all up, this game is very challenging and addictive and should be in the shops now.

The Home Filing Manager Temper Saver

by Derryck Croker

Remember Home Filing Manager? Yes, that disk that came free with 1050 disk drives? Did you consign it to the loft because you couldn't get a printout without italic characters? Did you wish that you could have more control over what printout you could get from it? Well I'd like to introduce you to The Home Filing Manager Temper Saver (HFMTS for short). For the sake of an hour or so's typing you will be able to tell HFM that you would like to print only a certain number of lines from cards, set the number of blank spaces between cards, and send printer control codes. All this can be achieved by simply adding a new card to your database! Multiple cards can be set up, and new options set or changed during the operation of the program. The only price to pay is that entries MUST be in hex. Table 1 shows hex and decimal equivalents for the range of values that can be used for the number of lines to print, and the number of spaces between cards. Refer to your printer manual for suitable hex printer codes.

Let's look at a typical HFMTS card entry. The chosen example sets HFM to print 7 lines (including the title line) from any card, print 2 blank lines, and sends to an Epson compatible printer the code to switch MSB control off. By a strange coincidence, this will set HFM up to print address lables, advancing the next label up to the print head ready for the next one, using normal typeface. The printer code tells the printer to ignore the high (MSB) bit of characters sent to it for printing. This bit is set by HFM to produce the black bar which descends the card during the printing process, and it is this bit which tells your printer to print italics. So here's the card:

MAILING LIST (title line)
0702 (line 1)
1B3D9B (line 2)

That's it! The title line, if used to describe the options contained, can be used to search for any particular HFMTS card, line one contains the number of lines to print, followed by the number of blank lines to print between cards. Line 2 contains the printer code already described. Notice that a) entries are 2 digit hex with no spaces between and b) the printer code string is terminated by a 9B, which is a carriage return. Omission of this digit may well result in incorrect response from the printer. The HFMTS program will allow one whole line of printer codes, including the 9B terminator.

Use the remainder of line one to describe the number of lines and spaces, useful since hex numbers are in use. HFMTS will trap most errors in your card's entries, and will not allow less than 1 or more than 18 lines to be used, since these lie beyond the boundaries of the card. No restrictions are placed on the number of blank spaces between cards, which can be between 0 and 255 ($FF hex).

Type in Listing 1 (hint, it's easier to photocopy the listing first, and cut it into manageable strips). Check it carefully and SAVE a copy before you run it, although there is a checksum routine built in which should trap most errors. Insert a formatted disk and follow the on-screen prompts carefully to produce your HFMTS disk. Do be careful. It's very easy to use your programming disk to write HFMTS to, and since it is a boot file it will overwrite DOS boot information and a portion of the first file on the disk. If you do make a mistake you will no longer be able to boot up DOS from that disk again, if you are fortunate the first file will be DOS, and so you may be able to rescue the remaining files to another disk. Unwanted or blank disks only please!

Warnings aside, and having produced a HFMTS disk, boot it up using whatever method is suitable to remove BASIC. Once it has booted up, insert your HFM master disk. Don't worry, your master disk will NOT be written to in any way, and in any case should be write protected. Press START, and HFMTS will check to ensure that the correct disk has been inserted. If not then insert the right disk and press START again. Should difficulties still ensue, then you should recheck your typing, or perhaps your HFM master disk is different to the ones on which HFMTS has been tested. Assuming, then,

Decimal values	Type this	
0	00	Not used for lines.
1	01	
2	02	
3	03	
4	04	
5	05	
6	06	
7	07	
8	08	
9	09	
10	0A	
11	0B	
12	0C	
13	0D	
14	0E	
15	0F	
16	10	
17	11	
18	12	
19	13	This is beyond the range for lines. It is not likely that you will need values beyond this range. If you do, then refer to a reference book. Values for printer codes should be in your printer's manual.

Table One : HEX Equivalents

that all is found OK by HFMTS, it will allow the remainder of HFM to load, and you should shortly be presented with the title screen. Once the program has fully loaded, insert the disk to which you would like to write your HFMTS card, and choose ADD from the menu line. And simply type in on the card the values that you would like to use! Remember though, that the title line should be used to describe the purpose of the card, which if using the example would be MAILING LABLES, line one of the card contains the number (in hex) of LINES and SPACES to print as 2 digits each (use leading zeros as appropriate), which should be typed in starting from the left margin. Line two of the card can be used to enter printer codes as per line one. Don't forget that you are limited to one line, and the codes should end with 9B. Should you not want to use printer control codes, then simply enter 9B at the left margin, and the printer will use whatever codes it recieved the last time. As a result of the terminator, the printer will produce a line feed. Now save the card. ALWAYS save a card before you use it.

It's time to set up HFM, so WITH YOUR CARD ON SCREEN, press S and RESET together. S stands for set! And thats it. When you print your next card, the printer will produce just the number of lines from the card, with just the number of blank lines that you requested.

```
1 REM ;***************************
2 REM ;* HFMTS LOADER & WRITER   *
3 REM ;*        FOR 8 16         *
4 REM ;*   BY DERRYCK CROKER     *
5 REM ;***************************
6 REM
10 REM *INITIALISE
20 RAM=20480:? CHR$(125) MSG=340:MSG2=
350:LINE=990
30 DIM IN$(1),DSK$(73),MSG$(28),MSG2$(
28),DAT$(91),HEX(22)
50 REM *HEX DATA ROUTINE ADAPTED FROM
ANALOG MAGAZINE
60 FOR N=0 TO 22 READ N HEX(N)=N NEXT
N FOR N=1 TO 73 READ
N DSK$(N,N)=CHR$(N) NEXT N
65 DATA 0,1,2,3,4,5,6,7,8,9,0,0,0,0,0,
0,0,10,11,12,13,14,15
70 DATA 104,169,49,141,0,3,169,1,141,1
,3,141,10,3,169,0,141,4,3,141,11,3,133
,213,189,5,133,203,160,87
80 DATA 141,2,3,169,80,141,5,3,32,83,2
28,48,24,238,10,3,173,4,3,24,105,128,1
41,4,3,173,5,3,105,0
90 DATA 141,5,3,198,203,208,227,173,3,
3,133,212,96
120 MSG$="Checking data    " POKE 752,1
 GOSUB MSG TRAP 200 RESTORE 1000
130 LINE=LINE+10 POSITION 12,4 ? "Line
 ",LINE READ DAT$:IF LEN(DAT$)<>90 TH
EN GOTO 230
135 REM *USE POS  FOR POSITION IN NEXT
 LINE
140 DATLIN=PEEK(183)+PEEK(184)*256 IF
DATLIN<>LINE THEN MSG$="Missing line "
 GOSUB MSG POSITION 18,5 ? LINE END
150 FOR X=1 TO 89 STEP 2 D1=ASC(DAT$(X
,X))-48 D2=ASC(DAT$(X+1,X+1))-48 BYTE=
HEX(D1)*16+HEX(D2)
160 IF PASS=2 THEN POKE RAM+N,BYTE N=N
+1 NEXT X READ CHKSUM GOTO 130
170 TOTAL=TOTAL+BYTE IF TOTAL>999 THEN
 TOTAL=TOTAL-1000
180 NEXT X READ CHKSUM IF TOTAL=CHKSUM
 THEN GOTO 130
190 GOTO 230
200 IF PEEK(195)<>6 THEN GOTO 230
210 IF PASS=0 THEN MSG$="Poking data
  " GOSUB
MSG PASS=2 LINE=990 N=0 RESTORE 1000 T
RAP 200 GOTO 130
220 GOTO 250
230 MSG$="Bad data-        " GOSUB MSG
POSITION 2,6 POKE 752,0 LIST LINE END
240 REM *SAVE HFMTS TO DISK
250 TRAP 40000 ? CHR$(125) MSG$="Inser
t destination disk" MSG2$="Press START
 to write HFMTS" GOSUB MSG GOSUB MSG2
260 IF PEEK(53279)<>6 THEN 260
265 ? CHR$(125) MSG$="Now writing
  " GOSUB MSG DISK=USR(ADR(DSK$)) REM
*WRITE HFMTS
280 REM *CHECK DISK STATUS
290 IF DISK=1 THEN GOTO 320 REM *DISK
OK
300 MSG$="Error in writing disk!" MSG2
$="Check disk and press RETURN" GOSUB
MSG:GOSUB MSG2
310 INPUT #16,IN$ GOTO 250
320 ? CHR$(125) MSG$="HFMTS disk OK!"
GOSUB MSG
330 GOTO 330
340 POSITION 5,5 ? MSG$ RETURN
350 POSITION 5,6 ? MSG2$ RETURN
1000 DATA 0005005006300993085C0A95185CC
A0270IC0915A0810F9A01F00C90600F9A90080
0403A93A0D0503A901800A03,694
1010 DATA A90080003A9030000502053E4EE
0A03A0D40318869080004O3A00503690O000503
CE005000E4A0008916380987,533
1020 DATA 5100A9801OF5A900A027915088I0
F0A94C60643AA97A80653AA95080663A1060A9
7A650CA950805O0A5D800034C,408
1030 DATA F614A208A0520EE6508CE750A212
A0D4A0FC02C93EFD098EE631BCE0314CF614AE
2A10AC201020405FF016C913,677
1040 DATA 8012B0E631AE2C10AC201020405I
8OE0314CC8504C7351A952850OA9108501A000
8100AAC0904008100A8204051,3
1050 DATA AA60A80A800852C998F000EEE650
0003EEE750C8000EA270A903904203A9089O4A
03A9009O4803A9909O4403A9,559
1060 DATA 5190450320S6E43029A909904203
A908904403A952904503A9FF904803A9009049
032056E43000A90C90420320,498
1070 DATA 56E44CF614A9E385CBA95185CC4C
7851088A20505I0A0A0A0A85C8982050510SC8
60206A51C8219003I8690929,130
1080 DATA 0F60C9109005C92780016OA98085
C0A95185CCA0278IC0915080I0F94C8451434F
50595249174O54503A980DA9,188
1090 DATA EEF3E5F2F400A0A6A080E0E1F3F4
E5F280E4E9F3E080E1EEE400F0F2E5F3F30083
04A1020400000000000000A2E1,939
1100 DATA E400E0E5F000E4E9E7E9F400EFF2
80EFF5F400EFE600F2E1EEE7E58180808080080
8080808080808080F2E5EEF4,454
1110 DATA E5F280E4EFE5F300EEEFF400F2E5
F3F00EFEEE4818080808080808000000000000000
00000000000000000000000,701
```

Listing One : HFMTS Loader and Writer

and in just the typeface that you specified. Great! Since the remainder of the card (using our example) is not printed, you can print an address label to our Editor so that you can tell him what a great magazine 8 16 is, and use the remainder of the card to remind yourself of any pertinent information, for example your subscription number.

And what do I do if I want to print out the whole of the card? Just press RESET and HFM will be told to print using its default values. Read whatever card is required and print it normally. Want to reset the print options AND the printer? In that case you'll need another HFMTS card:

```
RESET
1204 ;default lines and spaces
1B409B ;Epson compatible printer reset code
```

Bear in mind that some printer control codes may not work together, which may cause a problem from one card to another, and so you may need to set up a RESET card something like the above to use in an intermediate stage.

The HFMTS program will alert you if it cannot confirm the presence of an HFM master disk when you are requested to insert it on first loading, in which case rectify matters and press START again, if it finds incorrect hex digits or the number of

```
0100 ;*****************************
0110 ;* The HFM Tapper Saver      *
0120 ;*       for 8:16            *
0130 ;*  by Derryck Croker        *
0140 ;*****************************
0150       OPT NO LIST,OBJ
0160 ;
0170 ;o fem equates
0180 ;
0190 HFM =   $14F6    ;HFM reset entry
0200 LINES = $31E6    ;# lines
0210 SPACES = $31E0   ;# spaces
0220 OPTION LINE = $102A ;card line 1
0230 PRINTER LINE = $1052 ;line 2
0240 ;
0250       *=  $5000
0260 ;
0270 ;boot header
0280 ;
0290        BYTE $00,$05,$00,$58
0300        BYTE $06,$3A
0310 ;
0320 ;boot continuation
0330 ;
0340 LOAD LOOP
0350       LDA # <OPENING MESSAGE
0360       STA $CB
0370       LDA # >OPENING MESSAGE
0380       STA $CC
0390       LDY #$27
0400 START MESSAGE
0410       LDA ($CB),Y ;poke opening
0420       STA ($5A),Y ;into screen mem
0430       DEY
0440       BPL START MESSAGE
0450 ;
0460 ;wait for START
0470 ;
0480 START BUTTON
0490       LDA $001F    ;get console
0500       CMP #$06     ;START?
0510       BNE START BUTTON ;no
0520 ;
0530 ;read in HFM's boot sectors
0540 ;
0550       LDA #$00     ;set up
0560       STA $0304    ;buffer
0570       LDA #$3A     ;addresses
0580       STA $0305    ;in OCB
0590       LDA #$01     ;and sector
0600       STA $030A    ;LO
0610       LDA #$00     ;HI
0620       STA $030B
0630       LDA #$03     ;set up to
0640       STA $5000    ;read 3 sectors
0650 READ LOOP
0660       JSR $E453    ;fetch sector
0670       INC $030A    ;next sector
0680       LDA $0304    ;increment
0690       CLC          ;buffer
0700       ADC #$80     ;addresses
0710       STA $0304
0720       LDA $0305
0730       ADC #$00
0740       STA $0305
0750       DEC $5000    ;all read?
0760       BNE READ LOOP ;no
0770 ;
0780 ;check for correct disk
0790 ;
0800       LDY #$0A
0810 CHECK
0820       LDA $3A16,Y ;HFM copyright
0830       CMP COPYRIGHT,Y
0840       BNE LOAD LOOP ;no, restart
0850       DEY
0860       BPL CHECK
0870 ;
0880 ;clear message area
0890 ;
0900       LDA #$00
0910       LDY #$27
0920 CLEAR
0930       STA ($58),Y
0940       DEY
0950       BPL CLEAR
0960 ;
0970 ;insert new RESET handler
0980 ;
0990       LDA #$4C     ;JMP instruction
1000       STA $3A64
1010       LDA # <NEW RESET
1020       STA $3A65    ;to our new
1030       LDA # >NEW RESET ;RESET
1040       STA $3A66    ;handler
1050       CLC          ;return control
1060       RTS          ;to HFM's loader
1070 ;
1080 ;the new reset handler
1090 ;
1100 NEW RESET
1110       LDA # <NEW RESET
1120       STA $0C
1130       LDA # >NEW RESET
1140       STA $00
1150       LDA $06      ;RESET flag
1160       BNE OUR PROG ;$FF if pressed
1170       JMP HFM      ;else run HFM
1180 ;
1190 OUR PROG
1200 ;
1210 ;reset the printer buffer
```

Listing Two : HFMTS Source Code

lines to print is beyond the range 1-18, in which case press RESET and return the appropriate card to the screen and edit as required. Another message is produced if HFMTS cannot communicate with the printer, again a press of the RESET key is required. Don't forget, RESET by itself returns HFM to using its default values for printing cards, S and RESET together uses the displayed card to set up new values

Listing 2 is the assembly language used to create the DATA statements contained in listing 1. You do not need to type it in, it is simply intended for the student. I won't give you a blow-by-blow account of its working, since that would probably take up the remainder of the magazine, as would an account of the disk boot process. Suffice it to say that with one or two tricks to insert HFMTS into HFM's RESET chain and allow HFM's boot continuation code to run to load the remainder of its program, the rest is straightforward. Note, though, that I have relied on the fact that the OS will already have set most of the DCB (Device Control Block) for me when it booted HFMTS before I use it to load HFM's boot sectors. The OS will have also set $02FC (last key press) to $FF and closed all IOCB's as a part of its own RESET program before transferring control to the booted program

Have fun with HFMTS. I hope that it brings additional power to what is already a very easy to use database.

```
1220 ,address and criginal values
1230 ,of lines and spaces to print
1240 ;
1250     LDX # <PRINT BUFF
1260     LDY # >PRINT BUFF
1270     STX STORE+1
1280     STY STORE+2
1290     LDX #$12      ,lines on card
1300     LDY #$04      ,between cards
1310     LDA $02FC     ,get key
1320     CMP #$3E      ,S?
1330     BEQ ALTER     ,yes!
1340     STX LINES     ,restore
1350     STY SPACES    ,defaults
1360     JMP MFM       ,and run MFM
1370 ;
1380 ,fetch options from card
1390 ;
1400 ALTER
1410 ;
1420 ,fetch # lines to print (1-18)
1430 ;
1440     LDX OPTION LINE ,MSB
1450     LDY OPTION LINE+1 ,LSB
1460     JSR SCAN2HEX ,convert to hex
1470     BEQ RANGE     ,0 is invalid
1480     CMP #$13      ,as is >18
1490     BCS RANGE     ,as shoe error
1500     STA LINES     ,nee valve
1510 ;
1520 ,fetch # spaces between cards
1530 ,(no range check)
1540 ;
1550     LDX OPTION LINE+2 ,MSB
1560     LDY OPTION LINE+3 ,LSB
1570     JSR SCAN2HEX ;to hex
1580     STA SPACES    ,and insert
1590     JMP OVER      ,avoid range JMP
1600 RANGE
1610     JMP RANGE,ERROR ;shoe error
1620 ;
1630 ,nex fetch printer codes
1640 ;
1650 OVER
1660 ;
1670 ;set pointers
1680 ;
1690     LDA # <PRINTER LINE
1700     STA $00
1710     LDA # >PRINTER LINE
1720     STA $01
1730     LDY #$00
1740 LOOP
1750     LDA ($00),Y ;fetch
1760     TAX           ,MSB
1770     INY           ,next byte
1780     TYA           ;save index
1790     PHA           ,on stack
1800     LDA ($00),Y ,fetch
1810     TAY           ,LSB
1820     JSR SCAN2HEX ,fetch hex
1830     TAX           ,save in x
1840     PLA           ,reacve
1850     TAY           ,index
1860     INA           ,and hex
1870 STORE
1880     STA PRINT BUFF ,into buffer
1890     CMP #$9B      ,CR marks end
1900     BEQ FINISH    ,found
1910     INC STORE+1 ,next buff addr
1920     BNE NEXT
1930     INC STORE+2
1940 NEXT
1950     INY           ,next code
1960     BNE LOOP      ,back for more
1970 ;
1980 ,send printer codes   We need
1990 ,to close printer at start,
2000 ;since the OS RESET routine
2010 ,takes care of it for us
2020 ;
2030 FINISH
2040 ;
2050 ,oper #7,8,0,"P "
2060 ;
2070     LDX #$70      ,IOCB #7
2080     LDA #$03      ,open
2090     STA $0342,X
2100     LDA #$08      ,for write
2110     STA $034A,X
2120     LDA #$00
2130     STA $0348,X
2140     LDA # <PRINTER ;printer
2150     STA $0344,X ,device
2160     LDA # >PRINTER
2170     STA $0345,X
2180     JSR $E456     ,open it
2190     BMI PRINTER ERROR ,error
2200 ;
2210 ;send codes to printer
2220 ,using print #7,buffer
2230 ;
2240     LDA #$09      ,put text record
2250     STA $0342,X
2260     LDA # <PRINT BUFF ,buffer
2270     STA $0344,X ,address
2280     LDA # >PRINT BUFF
2290     STA $0345,X
2300     LDA #$FF      ,eox
2310     STA $0348,X ,255 chars
2320     LDA #$00
2330     STA $0349,X
```

```
2340      JSR $E456    ;print them
2350      BNE PRINTER.ERROR ;error
2360 ,
2370 ;close #7
2380 ,
2390      LDA #$0C     ;close
2400      STA $0342,X
2410      JSR $E456
2420 ,
2430 ;options set, run hfe
2440 ,
2450      JMP HFM
2460 ;
2470 ;printer errors trapped here
2480 ;
2490 PRINTER.ERROR
2500      LDA * <ERR.PRINTER
2510      STA $C8
2520      LDA * >ERR.PRINTER
2530      STA $CC
2540      JMP WRITE    ;write message
2550 ,
2560 ;converts 2 digit screen codes
2570 ;to a single hex byte returned
2580 ;in A.  X holds MSB, Y LSB
2590 ;only rudimentary checks made
2600 ;
2610 SCRN2HEX
2620      CLD          ;set binary mode
2630      TAX          ;get MSB
2640      JSR CONVERT
2650      ASL A        ;shift
2660      ASL A        ;to
2670      ASL A        ;MSB
2680      ASL A        ;position
2690      STA $C8      ;temp store
2700      TYA          ;new LSB
2710      JSR CONVERT
2720      ORA $C8      ;insert MSB
2730      RTS          ;finished
2740 ,
2750 CONVERT
2760      JSR RANGE.CHECK
2770      CMP #$21     ;A-F
2780      BCC 029      ;must be 0-9
2790      CLC          ;add
2800      ADC #$08     ;offset
2810 029
2820      AND #$0F     ;no upper nibble
2830      RTS
2840 ;
2850 RANGE.CHECK
2860      CMP #$10     ;less than 0?
2870      BCC RANGE.ERROR ;yes
2880      CMP #$27     ;more than F?
2890      BCS RANGE.ERROR ;yes
2900      RTS
2910 ,
2920 RANGE.ERROR
2930      LDA * <ERR.VALUE
2940      STA $C8
2950      LDA * >ERR.VALUE
2960      STA $CC
2970 ,
2980 ;writes error messages
2990 ,
3000 WRITE
3010      LDY #$27
3020 ERROR.LOOP
3030      LDA ($C8),Y
3040      STA ($58),Y
3050      DEY
3060      BPL ERROR.LOOP
3070 WAIT
3080      JMP WAIT     ;wait for RESET
3090 ;
3100 ;misc data and buffer, etc
3110 ,
3120 COPYRIGHT
3130       BYTE "COPYRIGHT"
3140 PRINTER
3150       BYTE "P ",$0B
3160 OPENING.MESSAGE
3170      .BYTE " Insert HFM master di
sk and press START "
3180 ERR.VALUE
3190      .BYTE "     Bad hex digit or
 out of range!     "
3200 ERR.PRINTER
3210      .BYTE "      Printer does n
ot respond!       "
3220 PRINT.BUFF
3230      *=  *
3240 ,
3250       END
```

Listing Two : HFMTS Source Code Continued

BaPAUG News

Its been a while since we reported on the Groups activities at the monthly meetings and I'm afraid that this issue isn't going to be much different. However, we have recently held our Annual General Meeting and feel that we should present a report for those members that did not attend.

It has been decided to increase the club membership fees. The new fees are £10 for adults and £5.80 for Juniors. Subscription rates to *8-16 are not set at the AGM and will change when and if required.

Currently three committee position have been filled, these being:

Chairman ..... Ian Brooker
Secretary ..... Mike Hosking
Treasurer ..... Colin Hunt

The group has also established an editorial team for *8-16, this being:

Editor ..... Colin Hunt
8bit Editor ..... Ian Brooker
ST Editor ..... Paul Brookes

How and who will control the groups PD libraries has still to be determined. However, the current plan is to have a PD Co-ordinator and assistants. Currently two ST user have volunteered to help.

Hopefully, full BaPAUG meeting reports will re-appear within the next issue of *8-16. Until then, happy Atari computing.

Colin

VCS Reviews

Double Dunk

Atari Corp £9.95
Reviewed by Rosemary Bach-Holzer & Thomas Holzer

Here is a game for all you basketball freaks. You've spent the day playing some friendly two-on-two basketball. You and your mate are psyched. You've aced the last three games, and the other guys are getting very desperate indeed. So! Let's play some serious basketball.

Insert the cartridge and on the screen appears two basketball players dancing to some funky music just to get you into the mood. Press "select" and the game option screen appears. The options are as follows: one or two players with the game length option between 2 to 15 minutes, which enables you to decide the length of time you wish to play or until one of the teams reaches a certain score which is between 10 to 48 points. You have the option to select the time or score you want to use. Three points can be awarded for shots made from the area outside the curved line, select yes or no or another option is if the offensive team does not shoot within 10 seconds, the other team gets the ball, select yes or no. The next option you may choose is a 3 second lane violation, again select yes or no, which means the other team gets the ball if an offensive player stays in the lane longer than 3 seconds at a time. If you then choose to penalise players for fouls then select foul detection, choose again yes or no and finally you now get a choice of six different team colours.

Having chosen from the above, press "fire" and start getting down to doing the "Harlem Globetrotters". There are two teams with two players called "Mr. Inside" and "Mr. Outside" competing in this basketball tournament.

You guide your players with the controller which means if your team has the ball, the controller moves to the player with the ball. If the other team has the ball, your controller moves your player who is guarding the person with the ball.

The players of each team have different playing skills, for example "Mr. Inside" is good at rebounding and blocking the shots, he can dunk his shots easier, and he sets picks for "Mr. Outside". He, on the other hand, is great at stealing the ball and making long shots.

Once the ball is in play, use the controller to move your players. When your team is on the offence, press "fire" once, which will then start the next part of your selected play. If you pull the joystick back and press 'fire' again whilst the ball is in play, the player with the ball will ignore the play and start a jump shot. Press "fire" once more whilst the player is in the air and he will throw the ball. If he is close enough to the basket he will then dunk the ball.

After an opponent misses a throw and you get the rebound you must clear the ball by moving your player until both feet are behind the 3-point line before you can shoot for a basket.

If you are on the defense, you can either try to steal the ball or jump to block a shot. To try to steal, press "fire" while your defensive player is close to the ball carrier and make sure he is dribbling the ball when the attempt is made. To block a shot wait until the other man jumps for the shot, but then you have to time your own jump with much accuracy in order to block the throw.

You score 2 points for a successful shot and if you choose the 3-point shot option, you score 3 points for shots made from the 3-point shot area. Shots from the foul line after a defensive foul are 1 point each.

The graphics of this cartridge are of good quality and the gameplay is better with a human partner because the vcs is very hard to beat. Then on the other hand, I must admit I am no basketball player so personally I would prefer playing with another person.

So, all in all it is an enjoyable game to play even for a rookie like me and also for the experts out there who are in the "Globetrotters" league.

Road Runner

Atari Corp £12.99
Reviewed By Thomas Holzer

I was told recently by a friend who works in the computer business that Atari UK is stopping the release of VCS and 7800 cartridges. So, I asked him what are the people going to buy, old carts only? If this is true, this is again a step in the wrong direction, because while I was flicking through some US console games magazines I saw full page advertisements from Atari promoting carts for the 7800 and 2600. I think the wrong people are working for Atari UK!!!

Anyway, he gave me the latest copy of Atari's Road Runner, so lets get on with the review.

Road Runner is a straight conversion from the coin-op of the same name and while it was not successful on the home computers, it looks very good on the VCS. The Road Runner is a scrolling type game from right to left in which you, as the Road Runner, run along a straight and twisted roads to escape the dab guy, Wile E. Coyote. Whilst running along you have to pick up bird seeds in order to gain speed and points. Sometimes a truck appears and you must dodge it otherwise you would be run over and therefore lose one of your three lives. A point to remember is to line up with the truck so that Wile does in fact get run over. But, the coyote is up to dirty tricks as well, because he replaces the bird seed with steel shot so, if you eat those he can catch you with a magnet. Or he rides on a rocket which makes him go faster than you and sometimes he can even shoot you. On later levels you encounter falling rocks, holes in the road (you have to jump over them) and land mines which you have to avoid.

The graphics for this game are first rate, with the Road Runner's legs going round in circles whilst he runs and his little hairdo going berserk in the wind. The background graphics consist of a desert with road signs, mountains and cactus, and parallax scrolling is used, even if it is a little on the jerky side. On the second side we have the title music from the cartoon (but don't quote me on this) including the famous "beep beep", otherwise there is only the typical Star Raiders whoosh for the running movements which might make you reach for the volume control after a while.

Overall, a new'ish (1989) cartridge for the VCS full of playability and cartoon-like fun. Get it while you can.

MIDI In Practice

By Jack Bartley

Whilst rehearsing for the Bournemouth Theatrical Promotions' 1991 production of 'Oliver' at the Pavilion Theatre, considerable difficulty was experienced by members of the chorus participating in the street-sellers scene in Act 2, where the five sellers combine in the 'Who will buy' number, each singing a different tune and each requiring to time their various entries into the song precisely to maintain the harmony. Quite frequently when several parts are sung in a chorus number the accompanist will make a tape for each of the singers to practice at home, as the one evening a week usually devoted to rehearsals is insufficient to attain a very good standard. With five parts plus the soloist as in 'Who will Buy' this becomes rather difficult to produce with all the parts correctly played and preserving the correct timings.

There is one way of producing a tape that is 100% correct in pitch, individual timing and relative timing and that is by using a Sequencer. Each part can be played using a separate track, either from the musical instrument keyboard in real time or step time with subsequent editing, or by punching in each note from the computer keyboard or the mouse in conjunction with the icons of the computer program graphics. The separate tracks can then be allocated different MIDI channel numbers and given individual MIDI programme numbers (patches) to play the track on any MIDI capable instrument. The instrument can subsequently be activated for cassette tape recording as and when each may be required.

So much for the theory of the requirement and now to get down to the practicalities of it, suitably armed with a copy of Lional Bart's score of the musical.

Loading my 520ST with the nearest thing I possessed to a sequencer at the time (albeit both Atari and PC versions) namely 'Music Studio'. Removing the program disk and inserting my music files disk I opened up a file called 'Oliver', went to options on the menu and clicked on 'Use Tracks' thus giving the capability of having music in 4 different tracks that could be replayed either individually or in any combination of the 4. Back again then to Options clicking on 'MIDI Parameters' and set up tracks 1 to 4, naming them after the sellers in the scene i.e. 1. Roses, 2. Milkmaid, 3. Strawbery (yes, I know it should be two b's, but not in the Music Studio track names list) and 4. Knives (for the knife grinders). Since the 5th singer - The Long Song Seller - (that's me folks) only contributes 3 notes at the end I considered that I could easily insert these from the organ keyboard onto the tape in the final mix with little fear of error. In the Channels column I allocated the tracks to Ch. 1 through 4. I did not allocate the 'preset' column at this stage - ain't it confusing, by 'Presets' I also mean 'Patches' by some sources or 'Program Numbers' by others.

Now to write the music from the score into the program. This version of Music Studio does not have a real time or step time capability so it's Hobsons choice to punch it in with the mouse from the icons. This is no real hardship as the song only consists of about 50 bars and some of the stanzas are repeated and only need placing on the staff once, and copying and pasting where they re-occur. Being in the track mode I am able to write all the parts for the separate instruments that are to represent the individual sellers parts on the same staff thereby ensuring that the relative timing can be accurate. At this stage I completely changed the batting order in the tracks screen in order to ensure that all the different colours for the parts were clearly distinguished on the staff. I then punched the whole lot in from the score, starting with the Rose seller, who is the first to sing, and then each of the others in the order of their entry into the song.

As I did not intend to use the Atari instrument for play back I used the internal sound purely for monitoring each tune, playing each back and checking it against the score, editing where necessary. For play back I turned to my Kawai Synth module K1m, selected on it my first multi (A.1) channel and programmed it with the 4 different single instruments which by choices happened to be internal A6, A8, B4 and B8 respectively. I was now able to set up the 'presets' on my 'MIDI parameters' screen, the patch numbers interpreting to numbers 6, 8, 12 and 16 respectively. All that remained to be done was to plug the Kawai output into a tape recorder, connect the Atari MIDI out to the K1 MIDI input, and play the individual parts from the Music Studio tracks screen separately and then in all different combinations, adding spoken explanatory commentary via the recorder microphone socket, so that each singer could hear their own part, practice it, and then turn to the section of tape where their own part was omitted and sing along with it.

Having completed this I then became a little more ambitious. The PC version of Music Studio has a capability of working in up to 8 tracks so I repeated the above exercise on the PC, anding the 5th track for the 'Long Song Seller', and a 6th track for the part of Oliver where the five sing with him. On the 7th track I keyed in a metronome, which on recording I put on the left stereo channel with the music on the right, thus enabling the singer to have it on for counting into his entry, and to mute it with the balance control when the timing had been mastered. If only the rest of the parts had been as simple to program as the metronome which consisted of one bar of 3 notes and the repeat programmed to 'n' times, the task would have taken much less than the numerous hours I spent on it. One doesn't have to work with a sequencer for long to realize why most of the pop songs today consist of one or two lines of melody and lyrics infinitely repeated - it's money for old rope!

The Basics Of A Sector

By Nir Darey (Acco, Israel)

[Re-printed from (FR)ANTIC March 1991 the newsletter of the Alamo Area Atari User Association (AAAUA)]

A disk can be formatted in different formats, Single Density, Dual Density (enhanced density) and Double Density. A sector is one of 18 sectors in a track at Single Density and one of 26 sectors in Enhanced Density. Because the track is a ring of magnetic disk (meaning that the track is circular), there is no way to distinguish between the beginning and the middle of a track. Therefore, a sector needs to be able to identify itself to the drive controller. To do this the sector has two parts: the header, and the data. The purpose of the sector header is to identify itself to the drive controller. The headers are written only when the diskette is being formatted. When reading or writing, the drive is automatically reading the headers to know which sector it is. The sector data is the contents that are being read or written to the sector.

The Read & Write Process

Whenever the computer performs a disk Write operation, the floppy disk controller requests each byte in turn and places it directly onto the surface of the diskette. The byte sequence is shown in the figure. As you can see, in the sector there are several spaces to protect the different contents.

Space 1: This space should be a least 1 byte long.
Space 2: This space must be 17 bytes long.
Space 3: This space should be over nine bytes long. This is to protect the next sector header from being over written.

When a sector is being read, the floppy disk controller is searching for the sector header. When the sector header is found, it compares the sector number and the track number to those given by the computer. If all is correct, the floppy disk controller begins searching for the data AM. If the data AM is found within 28 bytes the sector is read and transferred to the computer. Now the CRC status is checked for validity. If the sector is never found the processor will re-position the drive head in hope that the head had been on the wrong track.

The write process is identical to the read command, except that when the sector is found a write occurs.

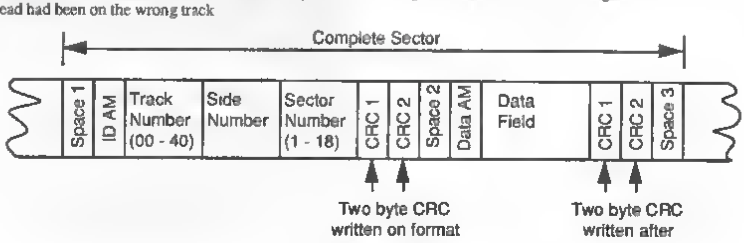
Sector Based Disk Protection

There are several different kinds of sectors used in protecting disks from being copied.

Bad or Missing Sector:	This sector occurs when junk data is written into the sector header.
CRC Error Sector:	This sector occurs when the CRC status at the end of the sector is checked. When these bytes do not match with the data that is being read from the sector, a CRC error will occur.
Double Sector:	This occurs when two or more sectors have be same ID number. If you read the sector you will get different sets of data.
Short Sector:	A short sector is a sector with less than 128 bytes.
Fuzzy or Phantom Sector:	When reading this sector you will get a CRC error and random data set.
Deleted Data Mark Sector:	This sector is a good sector but the data type flag has been changed to reflect a bad data mark.

It is almost impossible to control and change the sector header with an un-modified disk drive.

Well, that's all that I can say about the sector. Please forgive me if I missed something.



British SpartaDOS Time

Get the SpartaDOS Real Time Clock to Keep Proper Time and Fix the Date Notation into the Bargain

The Problem

As an otherwise firm convert to the advantages of SpartaDOS over the other varieties of DOS available on the 8-bit Atari, I have always been some what annoyed that the internal real-time clock runs slow - losing 10 minutes an hour in normal use. This criticism was also made by John S Davison in his review of SpartaDOS back in Issue 32 of Page 6 magazine and, as he suggested, the problem is due to the built-in clock being designed to run in the USA with a 60Hz VBI. Operating in the UK, or anywhere else, with a PAL system and 50Hz VBI makes the clock 'tick' that much more slowly and hence lose time.

For the rich amongst us an instant solution is to purchase the ICD R-Time 8 cartridge which runs its own internal hardware clock independently of the VBI. However, for users of SpartaDOS Version 3.2d on the XL/XE machines, there is a much cheaper (and easy) fix. Read on to save yourself some money.

The Method

Let's start with a little background information. The SpartaDOS time-of-day clock uses the standard Atari real-time clock register RTCLOK at locations 18, 19, and 20 (hex $12 to $14). These are incremented during every Immediate VBI, ie. 50 times a second, by the standard Atari Operating System. The accumulated count is read periodically by a SpartaDOS routine which uses it to update SpartaDOS's own internal clock and calendar registers (TIMER and DATER - see the SpartaDOS Construction Set Manual p.110). The latter are then used when putting time/date stamps on files, and by the TDLINE utility in displaying current time and date at the top of the screen.

The SpartaDOS update routine uses four built-in constants to convert the 'jiffy' count in RTCLOK to seconds, minutes, hours, and days. These are, as you might expect, based on the USA standard of 60 jiffies per second, ie. -

$00003C = 60 jiffies/sec
$000E10 = 3600 jiffies/min
$034BC0 = 216000 jiffies/hr
$4F1A00 = 5184000 jiffies/day

To correct the time keeping for British and other PAL systems running at 50 jiffies per second we need a different set of constants as follows -

$000032 = 50 jiffies/sec
$000BB8 = 3000 jiffies/min
$02BF20 = 180000 jiffies/hr
$41EB00 = 4320000 jiffies/day

- and some means of inserting these numbers into the SpartaDOS code.

The Solution

To make things difficult the relevant routine is neatly tucked away in the RAM underlying the OS ROM. My initial attempts to patch this code in situ, after loading SpartaDOS, were not very successful - usually causing a lock-up or crash.

The solution finally adopted is much simpler and works by patching the SpartaDOS X32D DOS file directly on disc. The amended DOS file can then be used to boot the computer as normal - but now with a clock running at the correct speed.

When you use TDLINE with this fix you will notice that the seconds on the displayed time increment a little jerkily. TDLINE is intended to update its display line every half-second, which it does by counting 30 VBIs. For UK use this count needs to be changed to 25 by altering a single byte in the TDLINE.COM file from $1E to $19.

While we are busy patching the time-keeping parameters it is also quite easy to dispose of the odd American custom of handling dates in a Month-Day-Year format, and change to our more familiar form of Day-Month-Year. What makes this particular fix slightly absurd is that SpartaDOS actually stores the date internally (and in its on-disc time/date stamps) in DD-MM-YY format already. The input and output routines which handle the date, swap the DD and MM values over specially to suit our American cousins. We can 'un-swap' the date values by interchanging two pairs of bytes in X32D.DOS - the bytes are part of temporary memory addresses used by SpartaDOS in its date manipulations.

The Program

The short BASIC program shown performs all of the necessary patches for you. It will patch any SpartaDOS disc containing X32D.DOS and TDLINE.COM. For those interested, the SpartaDOS DUMP command can be used to examine the contents of X32D.DOS and TDLINE.COM before and after the operation.

The revised constant values are patched into the X32D.DOS file, at file address $253B, and the swap of byte values to change the date format occurs between file addresses $1B8D/$1B9B for output, and $3060/$3066 for input. The single patch to TDLINE.COM is made at file address $036D.

Remember to UNLOCK the copy disc and UNPROTECT the files to be patched before running the

Caution	Do NOT attempt this patch operation on your original SpartaDOS Master disc - or even your usual working copy. Make a further copy and experiment using this until you are sure that everything is working properly. YOU HAVE BEEN WARNED!

By Terry Chamberlain

program - then follow the prompt.

After patching X32D DOS and TDLINE COM the only change in using SpartaDOS involves the DATE command. You will see that the current date is displayed in DD-MM-YY format, and you should enter the new date in the same way. When you display a directory of any SpartaDOS disc you will also see that the time/date stamps are shown in DD-MM-YY format.

A Final Note

Because the internal time-of-day clock is driven by the VBI, any program which disables the VBI, or interferes with the Immediate VBI vector, will stop the clock. Inhibiting Deferred VBI processing (as happens during I/O operations) only prevents update of the SpartaDOS time registers - RTCLOK keeps on counting in the background. When Deferred VBI processing resumes again the contents of RTCLOK are used to update the current time (that's what the four constants are used for).

And before anyone leaps to protest, I have to admit that the clock still runs slightly slow by about 10 seconds per hour. The accuracy is set directly by the frequency of the CPU clock oscillator fitted to the XL/XE PAL machines (3.546894MHz) and hardware counters within the ANTIC chip, and cannot be adjusted through any software fix. If you're not happy with this you'll just have to go out and buy that R-Time 8 cartridge after all.

```
100 REM ****************************
110 REM *                          *
120 REM *  SPARTADOS TIME AND DATE  *
130 REM *                          *
140 REM *        ADJUSTMENTS       *
150 REM *                          *
160 REM ****************************
170 REM * Terry Chamberlain - Apr90 *
180 REM ****************************
190 REM
200 Z=0 DIM D$(12),F$(15) F$(1,3)="D1:
"
210 ? ,? ,? "Insert SpartaDOS Disc In
Drive 1"
220 ?  ? "  Press <RETURN> When Ready
  " ?
230 CLOSE #1 OPEN #1,4,0,"K:"
240 GET #1,A IF A<>155 THEN 240
250 READ D$,A IF D$="DONE" THEN 300
260 F$(4)=D$ ? D$
270 CLOSE #1 OPEN #1,9,0,F$ POINT #1,A
,Z
280 READ A IF A<>-1 THEN PUT #1,A GOTO
 280
290 GOTO 250
300 CLOSE #1 END
310 DATA X320 DOS,9533,50,0,11,184,2,1
91,32,65, 235,0,-1
320 DATA X320 DOS,7053,250,-1
330 DATA X320 DOS,7067,251,-1
340 DATA X320 DOS,12304,74,-1
350 DATA X320 DOS,12390,75,-1
360 DATA TDLINE COM,871,25,-1
370 DATA DONE,-1
```

8-Bit Review

Desert Falcon (Atari Corp)
Cartridge RRP: £12.99
Reviewed by Thomas Holzer

All around you are long stretches of sand, ancient pyramids baking in the hot dead air and as always—danger. The legends that brought you here are 30 centuries old!! Hold on a minute, I remember a story beginning like this?? Oh yes, I reviewed DESERT FALCON in issue 5 of 8.16 for the VCS 2600. It didn't score high enough to really recommend it so let's see and find out about the 8-bit version.

For those of you who are not familiar with the plot here is the story. You play the part of a bird (the feathered kind) in search for treasure and fame and you have to fly through the desert in similar "ZAXXON" style scrolling and shoot everything in sight and also collect treasure by hopping or hovering over them. Collect three treasures for super powers like "smart bombs", extra speed and so on. To activate your super powers you have to press the joystick button twice but because you are always pressing "fire" anyway it really makes not a lot of difference. The enemy not only appears from under the desert sand (nice animation here) but also in flight and as usual the more you shoot the more points you earn. At the end of each level you will encounter a "sphinx", shoot it between the eyes several times to kill it. The next round will be a bonus round, so be sure to collect lots of treasure within the time limit.

The graphics of the "DESERT FALCON" are small but very good indeed, I like the way the bird is flapping it's wings, it is quite "royal" and realistic. The VCS "DESERT FALCON" is no match here. Some of the enemy sprites are a bit of a downer, many of them are just triangles and at the end of the level they seem to be flying under your shadow, slightly sloppy programming here.

Again, the colours of the desert are a mite peculiar, but not as bad as the VCS version and as for the sound all you get is a grating title tune and your average "shoot-bang-bang" FX.

Other than that it is quite an enjoyable shoot 'em up and you might want to buy it to add to your collection. All around you are long stretches of sand ... and so the story goes on

Russian Multiplication

By Simon Trew

Russian Multiplication is a system of multiplication which at first seems quite verbose but offers considerable advantages over our normal multiplication system.

How To Perform Russian Multiplication

The best way to describe it is to demonstrate. In Russian Multiplication, the method to multiply 19 by 17 (say) is as follows:

a) The two numbers are written down side by side, and the left-hand number is halved and the right-hand one is doubled. The results of this are placed on the line below and the process repeats until the left-hand number is equal to one (when halving, any fraction is ignored). Thus:

```
19 x  17
 9 x  34
 4 x  68
 2 x 136
 1 x 272
```

b) The lines where the left-hand number is even are crossed out, and the remaining numbers in the right-hand column are added to give the result:

```
19 x  17
 9 x  34
~~4 x  68~~  (strike out. 4 is even)
~~2 x 136~~  (strike out. 2 is even)
 1 x 272
--------+
      323 = 17 + 34 + 272
```

This method looks long winded but in fact is quite a useful method of mental arithmetic, as it has two advantages over standard multiplication. You only need to be able to add numbers together (to multiply a number by 2 you just add it to itself), and you only need to remember three numbers: the values of the two columns at each iteration, and the running total i.e. the total of all the preceding columns. With normal multiplication at least four numbers must be remembered.

```
     10001    (17 decimal)
   x 10011    (19 decimal)
   --------
     10001    (17 decimal)
    10001     (34 decimal)
   0
  0
 10001
00000
---------
100010000     (272 decimal)

101000011     (323 decimal)
---------
```

Example One

How Russian Multiplication Works

The Russian method can be seen fully when it is written out as a standard multupy but using the binary base, see Example 1.

It is obvious that whenever the top number (17) is to be multiplied is zero, then the result is zero. When multiplying by one, the result is the same as the top number (17) but with zeros to its right (just as when using decimal). This is in fact exactly what the Russian method does, as you cross out all the values that correspond to the multiples by zero, and the rest you add up, as they are already shifted.

What's The Point?

The Russian method also shows a very convenient (although hardly original) way of writing a multiply routine in machine code. To multiply 'b' by 'c', just shift 'b' to the left (multiply it by two) and shift 'c' to the right (divide it by two), adding the value of 'b' to a running total if 'c' is odd (its least significant bit is one).

In the Russian method, this stops when 'c' reaches one but it can of course be guaranteed that for any number this will not take more shifts than there are digits in the binary number. For example, if we do 16-bit multiplication then 'c' will reach 1 after 16 shifts at most. Obviously if it reaches 1 beforehand then any subsequent shifts make 'c' equal zero, this does not affect the result of the computation of course since 'c' is not odd so the value of 'b' is not added to the running total. The pseudo code for these two approaches are.

For the 'check zero' method

```
total = 0
while c > 0
  if c is odd then
    add b to total
  endif
  shift c right
  shift b left
endwhile
```

For the '16 regardless' method

```
total = 0
for loop = 1 to 16
  if c is odd then
    add b to total
  endif
  shift c right
  shift b left
next
```

Personally I opt for the '16 regardless' method for the Atari 8-bit. It is otherwise difficult to test that 'c' has reached zero. (This does make the routine a little less efficient since it will be performing needless cycles for some numbers.) A machine code algorithm for this method can be contructed as detailed

within listing one.

If you set the FACTB and FACTC words to two 16-bit numbers, then call MULTIPLY, the result of the multiplication will be placed in TOTAL.

Limitations

Here I have only allowed a 16-bit value for the result. The result could actually be up to 32 bits, but FACTB would also need to be 32 bits wide since it is shifted left, and more ADC statements would be needed between lines 1200 and 1210 to handle the other parts of the result. As it stands, if the result is greater than 65536 then you will only get the least significant 16 bits.

You could add the following lines so that if the result is greater than 65535 then OVERFLOW is nonzero (otherwise it is zero).

```
1025 OVERFLOW  BYTE 0
1085  STA OVERFLOW       , No overflow yet
1200  BCC NOLEVEN        , If too big
1209  INC OVERFLOW       , then overflow
```

Well, that's all there is to it. Next time it will be integer division. This is a bit more difficult but I shall try to dig up another defunct way of manipulating numbers. I'm still trying to work out how the Romans multiplied CDIX by MCCIV!

```
1000 FACTB  WORD 0    ; One factor
1010 FACTC  WORD 0    ; And the other
1020 TOTAL  WORD 0    ; The result
1030 BITS = 16        ; 16-bit arithmetic
1040 ;
1050 MULTIPLY
1060  LDA #0          ; Zero the total
1070  STA TOTAL       ; first! Load Y
1080  STA TOTAL+1     ; with the number
1090  LDY #BITS       ; of digits
1100 MULLOOP
1110  LDA FACTC       ; Check if C is
1120  LSR A           ; odd by calling
1130  BCC NOLEVEN     ; the carry bit
1140  CLC             ; If it is odd,
1150  LDA FACTA       ; add the value
1160  ADC TOTAL       ; of A to the
1170  STA TOTAL       ; total
1180  LDA FACTA+1
1190  ADC TOTAL+1
1200  STA TOTAL+1
1210 NOLEVEN
1220  LSR FACTC+1     ; Shift C right,
1230  ROR FACTC
1240  ASL FACTA       ; and shift 0 left
1250  ROL FACTB+1
1260  DEY             ; Check if we have
1270  BNE MULLOOP     ; done 16 cycles
1280  RTS             ; Return to prog

Listing One
```

Software Roundup

By Thomas Holzer

Atari 8-bit

Cavernia (Zeppelin Games)
Very good Platform romp. Price 3.99 and well worth it!!

DIGIDrum II (Grabn International)
Re-release of the 2-Bit Drum Synth that uses Replay sound samples

Atari VCS

Very quiet indeed. No new titles!

Atari 7800

Even more quiet right now

Lynx

Ultimate Chess Challenge (Telegames)
Qix (Telegames) ..... Can't wait to play that one
RC Destruction (Telegames)
Krazy Ace Golf (Telegames) ..... Sounds good
Strider 2 (US GOLD) ..... Platform hack
Crystal Mines 2 (Color Dreams)
Superskweeks (Lonciel) ..... A bit like skweek, the old ST game
Robotron (Shadowsoft) ..... Very old XL classic
Joust (Shadowsoft) ..... Another classic
Lemmings (??)... ... Can't remember the company, don't like the game anyway.
Scrapyard Dog (Atari Corp) ..... This could be the 7800 conversion
A.P.B. (Atari Corp) Good graphics and great speech

and more....

Xybots, Casino, Hard Drivin', Grid Runner, Toki, Viking Child, Bill and Ted, Cyberball, NFL, Hockey, Rolling Thunder, Geo Duel, Chequered Flag, Golf, Indiana, Soccer, Stun Runner, Vindicators, Turbo Sub, Hydra, 720, Basketbrawl, Pacland, Ishido, Basketball, Cabal and finally, **Shadow of the Beast**

All should be priced at £29.99. So, get your overdraft approved and look forward for some great entertainment. And remember: Anything the game boy can do, the LYNX can do as well, only better, bigger and in colour!!!

Portfolio

Swift BASIC (DIP Systems, 0483 301017)
Long awaited BASIC interpreter.

ST

Hanna Barbera Cartoon Character Collection (Hi Tec Software Limited)
Four games each featuring Top Cat, Yogi Bear, Ruff & Ready and Hong Kong Phooey.
Elf (Ocean)
+ many, many more too numerous to mention....

Fun with Stereo Sound on the TT030 & STE

By David Troy (C) 1991

You may guess from the title that I have a TT030 Much joy—it is true My impressions of the machine are favorable I can't tell you every program it won't run, nor every program it speeds up, simply because I use few programs I will tell you my adventures with this 68030 monster, and then we'll talk about something fun the much touted 8 bit stereo sound on the STE and TT

I received my TT030 in late May The machine was a 2MB machine with a 50MB hard disk. I had ordered an 8MB machine with an 80MB hard disk. It had no keyboard. And the extra RAM was included, it just wasn't installed This was all OK I figured I'd just use my Mega STE keyboard, install the RAM, and accept the fact that Atari wasn't going to use 80MB hard drives, only fifties

Before bothering to install the other RAM goodies, I thought I'd plug it in to make sure it worked. I plugged it into an Acer 7015 multisync monitor (hey, Jerry Pournelle, it works!), snapped the Mega STE keyboard in and turned it on It worked as expected. I took the TT apart and admired the swell chips inside. I found the place where the fast-RAM daughterboard plugs in It has four 1MB simms installed in it. That went in fine Then I located the ports where the 2MB of ST RAM installs I plugged in that board. I screwed everything back together and tested it once more Things looked promising, and then the screen went nuts

After much flailing around, I determined that the problem was the 2MB ST RAM board So I am currently in the process of getting it exchanged But Atari doesn't have any right now So I wait, and in the meantime, I have a 6MB TT030, with 2MB of ST RAM and 4MB of TT fast-RAM on that daughterboard. So, you're thinking, "Gee whiz, 6MB ought to be enough for even a psycho like Dave!" But there is a caveat.

There are two popular philosophies about computer design these days. The first, that which the ST was based around, says, "If we have a 68000 that runs at 8MHz then we can have everything on our address bus be based around 8MHz." This is handy It means that you can use a single hunk of RAM for the processor, Direct Memory Access (DMA), DMA sound, and video RAM DMA is cool It makes for fast laser printing and hard disk access Having the whole machine pinned to 8MHz also means that when you install an accelerator in your ST or use a 16MHz Mega STE, you're only speeding up the processor's internal operations, and whenever something on the address bus (any other chip, RAM, etc) must be reached, a giant switch is flipped, bringing the CPU back down to 8MHz

The second philosophy is to have a really fast CPU, like a 50MHz 68030 and give it some of its own RAM to play with You cut out stuff like DMA and DMA sound, and just leave a little bit of video dedicated RAM This is what the high and Macs do, and it's cool Since the Mac never had DMA, nobody's missing it there, and it runs super fast! (They don't have DMA laser printing, though!) The 68030 can use its own internal memory management unit (MMU) and talk to its ram in 68030 burst mode — allowing 32 bit RAM access! That is cool — very cool, and it's yet another reason why the TT can be faster than the ST Not only does it execute instructions four times faster, but it can grab at twice as much RAM per instruction!

So which philosophy did Atari use for the TT030? Exactly both The TT has one hunk of RAM it can use for the processor, DMA, DMA sound, and video RAM. That's the ST RAM It's the SLOW RAM. In the Mac IIfx, where DMA is not an issue, you can just have a little video RAM and use 030 fast-RAM for everything else. But Atari had to provide compatibility with all their DMA stuff. That means you have to keep some ST RAM, and a fair amount of it, too Want to provide compatibility with existing ST software? Might as well allow for 4MB of slow ST RAM But what about the 68030? It can do all that funky 32-bit addressing. It needs some RAM, too. That's the fast-RAM And it will only work with programs that don't need to do DMA, don't need to do DMA sound, and don't try to mess with video RAM directly Many programs do these things, and because of that, my 6MB isn't really 6MB

To set which programs should use the fast-RAM and which programs use the standard ST RAM, Atari made a program called PRGFLAGS, which allows the user to set bits in program headers which tell TOS what type of memory to use There are three user-selectable bits fastload, run-in-TT-RAM, and use-TT-RAM The fastload bit determines whether RAM should be cleared when a program loads. Clearing RAM takes time, so enabling the fastload bit prevents that clearing process. The run-in-TT-RAM bit determines whether a program is loaded into and executed from the fast-RAM Most programs can run in TT RAM unless they break too many rules.

Running a program in TT RAM may not save you too much ST RAM, though. Consider that your average program is about 300-400K in size What takes up most of your memory is the RAM your program allocates after it loads That's where the use-TT-RAM bit comes into play If it's set, any requests for memory will cause TT RAM to be issued This is great, provided that you're not trying to do any DMA stuff with that RAM. Enter *Calamus*

Calamus, the DTP program we all know and love, does swell things on a

TT with an SLM605 (or 804). Using it for the first time on a TT030 without any PRGFLAGS set, we discover, just as we discovered when using it on a Mega ST2, that 2MB of RAM isn't very much for *Calamus*. It's hard to use any decent fonts or have any graphics (and still be able to print it) when you've only got 2MB. To alleviate this, I turned on the fastload bit (for kicks) and the run-in-TT-RAM bit. This freed up about 300K of ST RAM, which left more room for our resident documents and DMA printer driver RAM. But still, it was a far cry from the 4MB machine I had been accustomed to.

So here I am, with a 6MB machine that is outperformed by a 4MB machine. For jollies, I turned on the use TT-RAM bit. It worked, and it had lots of RAM. I went to print. I got garbage. Yup, *Calamus* was using TT RAM, but it was printing a snapshot of the place in ST RAM where my page should have been. So I got garbage. So, is there a moral to this story? No. Just don't assume that more RAM is more RAM.

Calamus and the TT

Calamus has a few bugs on the TT. It crashes if you try to do anything that involves clicking on numbers to change values. For instance, suppose that I have some text I want in 55 points. Normally you'd click on the point size value, backspace over the old one, and then change it. (Since 55 points is not one of the 12 or so default values.) But Calamus crashes when you try this on the TT. Same thing happens when you try to change line spacing. But everything else works. I called Nathan at ISD and he said that *Calamus S* and *SL* will fix any TT compatibility problems and these programs should be out sometime this summer. Not only would *S* and *SL* work on the TT, but they would be rewritten and recompiled to take advantage of the 68030 and the built-in 68882 math coprocessor. (*Calamus* and *DynaCadd* supported the optional Mega SFP004 68881 chip but the 68881 in the Mega STE and the 68882 in the TT030 are at a different address.) So anyway, Calamus will be much happier very soon.

PageStream and the TT

PageStream works better on the TT at this point. While it doesn't run in the TT medium resolution, (neither does *Calamus*), it works well in the ST high rez mode. It also seems to be able to differentiate between ST RAM and TT RAM better. It can run from TT RAM and use TT RAM for documents and fonts, too. It just uses ST RAM for doing DMA laser printing. All in all, *PageStream* runs pretty fast on the TT.

Ultrascript and the TT

It seems that one way or another, every time I get a new machine, *Ultrascript* needs a song and a dance to run. I've mentioned I couldn't get version 2.1 to run on my Mega STE with the Extensible Control Panel installed. With the TT, the problems get deeper.

There are two problems. The SLM804/605 version has a typo in the address for the DMA port. This doesn't cause a problem on the ST, because it truncates the high end of addresses. But on the TT, that part of the address is valid and it points far off into never never land. *Ultrascript* also uses some self-modifying code which will not function correctly with the TT's processor cache activated.

So, Mike Fulton at Atari, who used to be with Neocept (*WordUp*), made a loader for Ultrascript on the TT. It does two things. First, it saves the current state of the cache, disables it, then finds the DMA typo in *Ultrascript*, fixes it, runs *Ultrascript*, then when done, restores the cache to its original state. I had a heck of a time getting it to work until recently, when I got a new version from Mike. It works on version 1.1 only, though, and Mike and I still don't know why it won't work on my machine with version 2.1. But that's ok. What is disappointing is that the program won't work with the TT's cache on, which is where much of your speed increase comes from. I wished that Imagen would re-do *Ultrascript* for the TT and take advantage of the 68030, its cache, and the 68882. But Imagen was purchased by QMS who set up an office in Alabama, letting go of most of the key people who wrote the program. I asked Mike if *Ultrascript* was really dead, and he said that there was still hope with Imagen. What that means, I don't know, but perhaps we will see *Ultrascript-030* sometime soon, or at least a comparable product.

Laser C and the TT

The *Laser C Shell* doesn't run on the TT. Megamax says that they will not be upgrading for the TT and that, "Turbo C has the market all sewn up in Europe, which is where the money is." So, that's that. But, while the shell doesn't run, the compiler and linker work just fine when run from a command line interpreter or other environment manager. And I have had a chance to write my first few programs in *Laser C* on the TT.

New Sounds on the Atari

When I got the Mega STE, I was curious how one went about programming the serial LAN (Appletalk compatible) port. Well, I couldn't try anything with it until I had another machine that had that port, and that was the TT. So I wrote a little program and was nearly successful in bopping bits back and forth on the LAN port. But for some reason or another, it wouldn't quite work. When I lost interest in that, I fell back on another project that I had started almost a year ago: playing ST Replay sounds through the STe/TT030 DMA sound hardware.

The 8-Bit Pulse Code Modulated Stereo Digitized Sound was a feature of the 1040STE that many people are still wondering about. Does anyone use it? How does it work? Is my keyclick stereo? Well, if you read the manuals on how it works, it's quite simple.

Hex Address	Name	Value
ff8900	Sound DMA Control	0 = Sound is OFF 1 = Play once 3 = Repeat forever
ff8902	Start Address (high)	leftmost 8 of 24 address bits
ff8904	Start Address (middle)	middle 8 of 24 address bits
ff8906	Start Address (low)	right 8 of 24 address bits
ff890e	End Address (high)	leftmost 8 of 24 address bits
ff8910	End Address (middle)	middle 8 of 24 address bits
ff8912	End Address (low)	right 8 of 24 address bits
ff8920	Sound Mode Control	Bits m000 00rr WHERE if m=0 Stereo, m=1 Mono rr decimal Sample Rate 00 0 6.258 KHz 01 1 12.517 KHz 10 2 25.033 KHz 11 3 50.066 KHz

Table W -- Common Addresses and their Origins

Essentially, all you have to do is get yourself a hunk of RAM (ST RAM), tell the computer where it is, and then tell it to start playing.

The program accompanying this article allows you to take one or two ST Replay format sound files and play them back in mono or stereo, respectively. The hardest problem to overcome here is that ST Replay saves files in an unsigned eight-bit format. What this means is that when the sound chip sees a 0, it will cause full negative displacement of the speaker, and when it sees a 255, it will cause full positive displacement. The intermediate 128 has the effect of not moving the speaker at all. This is cool.

The STE/TT hardware expects that the sound data be presented a little differently. Rather than having the unsigned data, with 0-255 as the range, the chip wants signed data, with a range of -127 to +128. Full negative displacement is -127 and full positive displacement is 128. Zero is now the intermediate value, and it has the same effect 128 has in ST Replay format, of not moving the speaker at all. So, we can see that we need to create a function that maps from the unsigned data to the signed data.

Atari's choice of mapping their signed values makes our function writing fairly easy. To the sound hardware, 0 through 127 really are 0 through 127, but 128 through 255 are really -128 through -1. So to go from ST Replay format to STE/TT format, we need to map the values 0-127 to 128-255, and map the values 128-255 to 0-127. This means, in effect, that all we have to do is take our data and reflect it around 127. If a value is greater than 127, subtract 128 from it. If it's less than 127, add 128 to it. This does all of our remapping.

When the sound hardware is in mono mode, it fetches one eight-bit byte at a time and plays it at the the sample speed through both the left and right sound ports. When the sound hardware is in stereo mode, it fetches a 16-bit word at a time, each word containing two bytes: the values for the right and left sound channel, the left channel being in the right half and the right channel in the left. (It makes sense if you think binary.)

If we read in two ST Replay format sound files to be played in stereo, we have to pair up each byte, one-to-one, left-to-right in memory before we can play them in stereo. That, too is a fairly simple and quick procedure. That is,

```
/* ST Replay format digitized sound player for TT/STE Hardcore
(C) 1991 David C. Troy - May not be reproduced without expressed
thought.  */

Include "portab.h" /* Standard Atari Headers.  Should be OK with */
Include "osbind.h" /* Laser, Bionic, Mark Williams, etc.  */

/* Macros to allow easy write/read to memory locations.  Set a
initial thought.  Translates to: THE 16-bit (UWORD) CONTENTS of the
MEMORY LOCATION 0xwhatever POINTED TO by the number 0xwhatever.  */

#define DMASCTRL (*((UBOBO *)0xff8900))
#define FRMBASEH (*((UWORD *)0xff8902))
#define FRMBASEM (*((UWORD *)0xff8904))
#define FRMBASEL (*((UWORD *)0xff8906))
#define FRMEND0 (*((UWORD *)0xff8907))
#define FRMEND1 (*((UWORD *)0xff8909))
#define FRMEND2 (*((UWORD *)0xff890B))
#define SNDMODE (*((UWORD *)0xff8920))

/* I'm using some global variables.  This is bad practice, but it
makes writing a little goofy progress like this easier.

   save-sup = user stack for supervisor code routines
   frase    = start address of left/combined data frase
   rfrase   = start address of right sound channel data frase
   reize    = size of right sound channel file
   reselze  = amount of RAM required for our left/combined
              data frase
   ch-size  = minimum of either leise or reize, just = to
              leize if mono
   Mono     = boolean to tell whether we re in mono or not
   fp       = 16-bit Geodos assterisk file pointer */

LONG save-sup,frase,rfrase,reize,reselze,ch-size,
BOOLEAN Mono,
UBBO in,

sup-on()    /* Turn on Supervisor Mode */
{
  save-sup = Super(0L);
}

sup-off()    /* Turn off Supervisor Mode */
{
  Super(save-sup);
}

Sign()    /* Converts from Replay unsigned to Atari signed format */
{
  LONG c=0,

  while (c++<=reselze)
  {
    if (*(unsigned char *)(frase+c) <0x80)
       *(unsigned char *)(frase+c) += 0x80,
    else
      *(unsigned char *)(frase+c) -= 0x80,
  }
}

Stereo()    /* Takes left and right data and combines into Stereo
Mode  */
{
  LONG c=ch-size,c2=0,
  while (c-- >= 0)
  {
    ((UBYTE *)frase)[reselze - c2++] = ((UBYTE *)frase)[c],
    ((UBYTE *)frase)[reselze - c2++] = ((UBYTE *)rfrase)[c],
  }
}

main()
{
  /* More variables

   lfnase  = Left Channel Filename
   rfnase  = Right Channel Filename
   S_or_M  = Stereo Or Mono? character variable
   speed   = SFad we're playing example at.  Range 0-3
   err     = Error flag to check validity of Fsfirst
   oldDta  = Your machine's default dta address.  We re saving it in
             oldDta and then changing it to needDta for our use
   needDta = Our new dta address.  Allows us easy access to
```

provided you've got enough RAM to have two buffers — one to hold the combined stereo data and one to hold the right channel by itself. If you try to combine the two using a single buffer that is just long enough to hold the combined stereo data, you end up doing a very long and tedious combination algorithm. (If anyone knows a quick way to do this, let me know.) The other way to do this is to read in the data from the disk files one byte at a time, but that too is slow, at least in C. Again, I do not claim to be "Master Programmer, Chief of the C Tribe," so if anyone has any better ideas on how to do what I've done, let me know. I'm trying to learn, too.

Just going through the program quickly, you'll notice the macros I set up to allow access to the DMA sound chip registers. They allow easy "poking" and "peeking" to those registers and are quite helpful.

The sup-on() and sup-off() routines deal with the 68xxx supervisory programming mode. Essentially, supervisor mode allows us to tweak the sound registers which live in a protected part of memory. So we turn on supervisor mode to get access to the sound registers and turn it off when we're done.

The Sign() and Stereo() routines perform the functions we already talked about, they convert to STE/TT format and then do the Stereo pairing up if necessary.

The main program is easy to follow, even if poorly structured. We start by setting up a data structure called "newdta," along with some other variables. The dta is a 44 byte structure that gets filled with information about a file (like its size, its creation date, filename, etc.) when you execute a Gemdos function called Fsfirst(). We use that to make sure we have enough memory to load your sample files without actually loading them. So we just tell Gemdos that we want to remember where the dta used to be, with the line olddta = Fgetdta(). Then we tell it to use our newdta by saying Fsetdta(newdta). From there we find out whether to do stereo or mono, what the filenames are, and what size they are.

If we're doing mono, we only need as much ram as the left channel file, but if we're doing stereo, we need twice as much ram as the smallest of the left and right channels. If that ram is available, then we read in the left channel and the right channel if it's stereo. If it's stereo, we do the Stereo() function and pair up

```
            file info
  ok        = Boolean to determine if Malloc() was successful
*/

char lfname[40],rfname[40],S-or-M,speed=f;
int err;
long olddta = Fgetdta();
struct {
  char    crud[25];
  long    fsize;
  char    fname[14];
} newdta;

BOOLEAN ok;

Fsetdta(&newdta);    /* Set dta address to our own newdta */

printf("[S]tereo or [M]ono Sample?\n");
scanf("%s",&S-or-M);          /* Is this Stereo or Mono */
if (tolower(S-or-M) ==  'm')
  Mono = TRUE
else
  Mono = FALSE;
err = 1;
while (err)      /* Keep on getting lfname until correct  */
{
  printf("Left Channel Sample Filename ");
  scanf("%s",lfname);
  err = Fsfirst(lfname,0);
}

/* The RAM we need  is the size of the left channel sample file
ch-size = the smaller of the left or right channel
At this point, it's the left channel */

newsize = ch-size = newdta.fsize;
printf("%s is %ld Bytes\n",lfname,ch-size);
if (!Mono)       /* If NOT Mono  it must be stereo  */
{
  err = 1;       /* Keep on getting rfname until correct */
  while(err)
  {

    printf("Right Channel Sample Filename  ");
    scanf("%s",rfname);
    err = Fsfirst(rfname,0);
  }
  /* The right channel size is the size of its file  */
  newsize = newdta.fsize;
  printf("%s is %ld Bytes\n",rfname,newsize);
  if (newsize<ch-size)  /* If newsize is less than ch-size is new, */
    ch-size = newsize;          /* then ch-size = newsize       */
  /* So, below we work with RAM as the smallest sound channel
  newsize = ch-size * 2;
}
/* Unless both of these Mallocs are OK, we can't execute the
   program.  */
ok = FALSE
ok = (lfree = (long)Malloc(newsize))>0;
if (!Mono)
  ok &= (rfree = (long)Malloc(ch-size))>0;
if (ok)
{
  /* Read in left sample file into the left/combined data frame */
  fp = Fopen(lfname,0);
  Fread(fp,ch-size,lfree),
  Fclose(fp)
  if (!Mono)     /* Then it must be stereo  --> */
  {
    /* So read in the right sample file into the right frame */
    fp = Fopen(rfname,0);
    Fread(fp,ch-size,rfree),
    Fclose(fp),
    Stereo();     /* Copy free rfree to free (combined) */
  }
  Sign();         /* Do the ST Reday -> STE/TT Conversion */
  /* DO THE SOUNDS */
  sup_on(),
  FRMBASEH = lfree>>16,     /* Tell it start address */
  FRMBASEM = lfree>>8,
  FRMBASEL = lfree,
  FRMENDH = (lfree+newsize)>>16;  /* Tell it end address */
  FRMENDM = (lfree+newsize)>>8,
  FRMENDL = (lfree+newsize),
```

the little buggerbytes. We convert from the ST Replay format to the STE/TT format with the Sign() function, turn on supervisor mode, set the sound registers and then sound's a playin' I put a little loop in there to let you change the speed it plays at. That's fun. Then, when you're done, you can leave the sound playing. That's kind of cool - you have sound playing while you work. Since I didn't allocate that RAM for my program, though, that RAM can and will be used by other programs which means that your sound will be replaced by the wonderful sound of *WordPerfect*, or whatever program you're using. That often sounds like a dying hyena or machine gun fire, or a bad joke. To disable that, either reset your machine or run my program again, this time exiting with the "AH! Save Me!" option.

I should mention that this program is set up to loop through your sound forever and ever, all without detriment to other computer functions. This is a feature of the DMA sound hardware and it seemed like a good idea to use it, so you could really get to know your sound file. What I did to test this program was use ST Replay and digitized a nine second wailing vocal from the song "Belong" on R.E.M.'s latest release Out of Time. It lends itself to repetition because they repeat it in the song. I digitized the left channel and then digitized the right channel, and tweaked them both until they looped perfectly. Running them through my program, I find I didn't get them exactly in sync, but I'm awful close and it gives a kind of "stadium" effect. Yes, you can tell it's stereo.

If you don't have ST Replay there are a ton of sound files that are compatible. Just get a couple and pay them back at the same time, in stereo! You can find them on GEnie or in public domain libraries. You may note that ST Replay does not digitize at any of the STE/TT sample rates mentioned in Table W. So, you will get some speedup or slowdown no matter what you do. To change the frequency of an ST Replay file would require some serious killer Calculus — a Fast Fourier Transform. I'm working on it. I'll get back to you. Other more complex techniques, like working with interrupts, are needed to string together different pieces of songs in various combinations. I just thought y'all might like to make some use of your DMA sound.

The information in the table probably belongs to Atari Corp. somehow or another.

```
    DMASCTRL = 3;                /* Repeate Forever */
    sup-off();
    Feeldle(oldidle),     /* Reeete dle to oldidle velue */
    while (speed<4)  /* Rlieee user selectable speed - just fun! */
    {
      sup-on();
      SHOMODE = (char)None << 7 | speed;
      sup-off();
      printf("\n\nO  8 25# KHz\n");
      printf("1  12 517 KHz\n");
      printf("2  25 033 KH2\n");
      printf("3  50 066 KHz\n");
      printf("4  Alee lhie pepelele etend - leeue neise running\n");
      printf("5  Alee lhie pepelele etend - AH! Seue Mel\n->");
      speed = Ceenin()-48;
    }
    if (speed==5)       /* Turn off sound before ee leeue  */
    {
      sup-on();
      DMASCTRL = 0;
      sup-off();
    }
    Mfree(freee),         /* Freee up ceebined freee */
    Mfree(rfreee),        /* Freee ep right freee */
    printf("\nSee ye\n");
  }
  else
    printf("Silly rabbit, telx are for kids! You don't have enough RAM!\n");
}
```

Listing One: Source Code

To use this program you'll need a C compiler. I'd love to hear which ones worked for this program. I had no trouble using *Laser C* (from a CLI) on the TT, and a compiled version of this program ran just fine on the TT, the 1040STE and MegaSTE. DO NOT attempt to run this program on a non-STE. It will probably crash your machine.

I suppose that's all I have room to say. If anyone has any questions or comments on my programming technique or algorithms, or has a good way to do a fast Fourier transform, let me know. And if anyone would like to try my program on their STE, but don't have a C compiler, I will post an ARC with everything in it on GEnie. And if you don't have a modem, send me $5 and I'll send you a disk. If you have any questions about the TT, get a hold of me in one of the usual ways.

COOL WAYS TO REACH DAVE.

Phone: (301) 544-6943
FAX: (301) 544-1FAX
MAIL: David Trox
556 Balt. Annapolis Blvd.
Severna Park, MD 21146
GENIE: Toad-Serv
CompuServe: 72470,1605
Internet: dtrojh@jhunix.hcf.jhu.edu

AtariWriter Plus VI

For Those Who Don't Want to Read the Book

By Jimmy Boyce (CACE)

Reprinted from Atari Interface Magazine, January 1990

Well folks, I got my AW+ book back and...no notes. So I will start with the following premise: we have completed our discussion on the dictionary and we have written, edited and corrected the spelling of our great literary masterpiece (oh yes, and saved it to disk).

Printing our Masterpiece

This leaves us with printing it for the eyes of the privileged and chosen (but not for the sneaks that boot up AW+ and get into our private file disks and already know what we are about to print) <pg15>.

Once again, this segment is menu driven and is really quite simple. However, there are some fun things we will get into later on, such as printer drivers. I will be devoting an entire segment to this nasty little subject and hopefully will clear up some of the fog in the book.

Don't Forget to Turn On Your Printer!

One of the first steps to successful printing is turning on the printer. Don't laugh, I wish I had a nickle for every time I forgot that little item. I notice in the book that it mentions that some printer interfaces have to be switched on, so if you have that to do - do it! Smash the [ESC] key (I own stock in Atari) and - viola - there on your CRT (fancy for monitor screen) is the main menu.

Indulge yourself and strike the [P] key for PRINT FILE. Down at the bottom of your screen is the burning question: PRINT TO PRINTER Y/N? Go ahead, be brave and strike [Y].

Using Atari Printers

Now the screen has a whole list of Atari printers listed, from the 825 to the XDM121. If you have any of the Atari printers listed on the screen, push the appropriate letter listed to the left of the equal sign and a program resident printer driver will be loaded for you.

This for Atari printer owners only - if your Atari printer is not listed try the letter [A]. If the printer isn't listed under the letter [A], you must re-boot AW+ and try the letter [F], usually one or the other will work for you.

Let's assume one of the two letters worked or that you have a listed Atari printer. The next question at the bottom of the screen will be PRINT WHOLE DOCUMENT, Y/N? Type in the letter [Y].

You are asked how many time you want this thing to be printed, it is your option, up to and including 99 copies of a single document. Usually, I ask for one and take it to a copier and get copies made. This saves a fair amount of wear and tear on the printer. At this point your printer will start printing - usually.

And If You Have Another Printer?

I will explain the pitfalls and problems later. Right now, I am going to continue on with a discussion of the letter [H] option on the printer menu.

Yes, that all inclusive letter [H] which stands for OTHER. Almost intimidating isn't it? It simply means that you are some sort of a low life that does not own an Atari printer (like me).

Epson Printers

Gently depress the [H] key and three printers are listed, the first being an Epson - "Oh, happy days," you say! "Mine is an Epson compatible." Try the letter [E] if this is the case, but don't be surprised if you do not get full use of your printer's capability, at least I could not.

What about the next one? I don't know, because I know nothing of the printer, the same applies to the last printer listed. And that leave us with the final choice - that an inclusive other. Go ahead - be chancy and see what happens, I dare you! All that happens is that another request appears at the bottom of the screen. It asks you to enter your printer driver disk filename.

My Printer's Not Listed!

Well, you do not have a printer driver, so what are you going to do? This is what you are going to do - you are going to wait until next month when we go through in detail setting up a custom printer driver for your individual printer.

This gives you a month to find that moldy old printer book that came with your printer or a chance to borrow one from someone else, because without it you just simply cannot enter the commands necessary for your printer.

Happy hunting. Bye now.

Wanted - Articles for future issues of 8:16

We are very short of articles and programs for future issues. Please help us to help you by submitting any articles on any computer related subject you may be interested in. Subscription increased by one for each of your articles published.

Supercharger, as is

The External PC Emulator For The ST

In this article, **Michael Nyman** provides us with an overview of PC emulation using Supercharger.

The reason I say "as is", is that since last May I have been waiting for a drastic upgrade to arrive from Germany, but as yet (end of July) it still has not arrived. This shows that the Germans are not necessarily more efficient than the British, in spite of all their hype.

Supercharger comes in the form of 2 boxes, one measuring about 6 x 7 x 2 inches which contains the brains and memory of the system and the other measuring 5 x 3 x 2.5 also in inches, which contains the power supply and runs hot as it is never disconnected.

On the rear of the main box is the power in socket and an on-off switch, plus a DMA in and a DMA out socket, while on the front is a reset button plus 2 light emitting diodes, one green to show the power is on, the other red which flashes when data is being transmitted to and from the main machine. The box plugs into the DMA socket on the Atari or the DMA out socket on a hard drive if one is in use.

The software comes on 3 double sided disks which need installing onto either a floppy work disk or the hard drive. The software does include DOS 4.01 and installation is fairly simple. On one occasion however after my hard drive went down I couldn't manage to get it running again without a 30 minute phone call to Eric Moore of Condor Computing (the UK suppliers of Supercharger). During the call Eric gave me step by step instructions and we were up and running again. Since then I have had no trouble.

The main box contains a NEC V30 processor which is a speeded up version of the 8086 (heart of the PC/XT) and 1 Meg of memory plus some other electronics which I don't need to know about.

When running, Supercharger emulates a PC/XT and will run any software that that machine will run and slightly faster. It uses the ST's keyboard, mouse and disk drives just as if they were on a PC and it will read ST disks as well as PCs. It also uses the ST's screen and can output either CGA colour or Hercules Mono. As I use an Atari Mono monitor I can only get it in either CGA or Hercules mono, both of which I find superior to their equivalent on an actual PC. In PC mode I cannot fault it except that sometimes I think it runs too fast or the keyboard is too fast for occasionally it crashes, causing the usual frustrations.

When the software is installed on the ST and it has been re-booted "Supercharger" comes up in the accessory file and clicking on it then following the on screen prompts gets one to the usual PC prompt of "A>" and from then on you are on your own. A hot key press gets you back to the ST and again clicking on the Supercharger accessory gets you into PC mode, where you left off. Condor sent me some 4 disks of PD software including a word processor "Galaxy" and two discs of DOS tutorials, so that gave a good start. The manual that comes with the system is comprehensive and in parts much too technical for me, it contains some 104 pages.

— being entirely external to the ST there is no trouble about connecting it up

One beauty of Supercharger is that being entirely external to the ST there is no trouble about connecting it up and being self-contained one can use the 1 Meg of RAM it contains as a RAMdisk when in ST mode and in theory one can use the ST's RAM as a RAMdisk when in PC mode. To do the latter one has to set up a reset proof RAMdisk in Atari mode before one goes into PC. The only problem is that the RAMdisk provided on the system disk does not work on my set up and during the long phone call mentioned above Eric was unable to find out why, so that is a facility I do not yet have.

I have been using Supercharger for just over a year now and have had much joy from it with the exception of the limitation just mentioned. I fine it an easy way to use PC programmes as used by thousands of others, but with one exception I have not found the PC able to do anything that cannot be done just as well, if not better, by the ST. The one exception is a Money Management programme for club accounts, for I do not find anything as good as "Money Manager" on the PC. Supercharger does enable me to share programmes and data with others.

One thing to watch out for is that because Supercharger will read from and write to Atari disks it is possible that one can forget oneself and work with ST formatted discs which will subsequently not work with a genuine PC and find one self cursed by others. Unfortunately that is their had luck and shows that the ST is more flexible that the PC.

As stated at the beginning of this review I am waiting for an upgrade which will contain an 80286 processor and 4 Meg of RAM, which I hope will give me one hell of a system.

Editorial Comment

By Paul Bronkes

The recent adverts for Supercharger SC plus 286 mention that an external PC cardframe enclosure with two 16-bit slots is now included. This will allow real PC expansion cards such as Super VGA Colour (1024 x 768 in 256 colours), PAL/EPROM programmers, A/D cards, Scanners etc to be used with Supercharger. This is extremely useful for people who want to make use of the thousands of PC add-ons available. The slot for a real SVGA card is the feature that I think will really tempt a lot of people.

In the American AIM magazine, there have been adverts mentioning that a 386 version of Supercharger is under development. This is especially needed for Microsoft Windows, as Windows only really becomes useful when run with a 386 processor. The 286 will run windows, but quite a lot of use is made of the hard disk to store away bits of memory. In fact, because of the memory limitation of DOS (640K directly), Windows applications tend to run out of memory very quickly. Some utilities are available to relocate parts of the DOS operating system, mouse drivers etc out of the 640K memory into extended memory which frees up a little more memory for Windows, but not much. The 386 mode of running windows is reported to be superior because it can use the expanded memory in your machine. In a real PC this could only be another 360K or it could be several megabytes depending on the amount of RAM fitted. The extended memory is used instead of the hard disk, thus increasing the speed at which Windows applications run.

Deskjet 500 Update

By Paul Brookes

300 DPI colour printing on a Deskjet/+/500

In the last issue I praised the Deskjet 500. Now it is even better value due to large discounts that are available. This printer is selling very well and as a result, there are some interesting developments centred around it.

The most notable of these are the colour printing kits that are becoming available. Colour printing using the DeskJets relies on using several coloured ink cartridges, one after the other. A three-colour page would be split into three monochrome pages. Each page would only contain the text and graphics for a single colour. The first page would be printed using the first colour. Then the printhead cartridge would be changed to the next colour and the monochrome page representing that colour would then be printed. Finally, the print cartridge would be changed to the third colour and the associated monochrome page sent to the printer. If this sounds too much bother, then you can always go out and buy a Paintjet XL!

A friend of mine, Mirko, has been printing in several colours on a DeskJet for over six months, refilling with cheap, non-smudge ink he bought in his native Germany. The method of refilling is a little crude however. Nurse! Where's that syringe..?

The impressive results, such as Mirko's seven colour printouts, make the colour refilling kits seem like a good investment. Now, all we need is a hand full of empty cartridges! That's about £80 worth of standard HP inkjet print heads.

Several companies are now advertising refill kits. One I saw recently, is supplying ready to use colour cartridges. Come on HP, halve the price of the cartridges and give us a choice of colours. Now wouldn't that be nice?

GDOS

Working Title, the publisher of the excellent **Calligrapher** package, have released a new DeskJet 500 GDOS driver. This makes use of the data compression algorithm employed by the latest DeskJet, to reduce data transfer times and ultimately reduce print times for A4 300 DPI graphics, which can take several minutes with some software.

ST34

This is the club's DeskJet/LaserJet utilities disk (available for £2 from the usual address) and it contains the following files:

P_OR_SAV

Despite its name, this is a great utility for DeskJet owners. It produces amazing screen dumps in two sizes and in draft or NLQ. There is also an EPSON version. Both versions will also save the screen to disk as a DEGAS picture. Written by Chet Walters.

EPSJET

Tries hard to make the DeskJet emulate an Epson MX. Great if you like Epson quality output! Works with GDOS too. Written by David L. Bailey.

DJ_RESET

A utility in ACC and PRG form, for DeskJet owners who find that they have to turn on their printer before their ST if they are to avoid locking the printer port up. This utility resets the ST's printer port if it locked up when powering on the DeskJet after the ST. It saves having to re-boot the ST unless you're stuck in a program like Signum. Written by yours truly.

JETSET

Allows some font and page selection commands to be sent from the ST via an accessory dialog box. Useful, but needs updating for DJ500 owners. Written by ????

DJBOOT

Automatically sets draft mode on the DeskJet when the program is run from the AUTO folder. (Not required on the DJ 500). Written by Doug Harrison.

BOOKER

This program produces A5 booklets from text files presented to it. It prints 2 pages side by side in landscape mode, on both sides of the paper (you must turn it over of course!). Written by Sol Guber.

The 34th Longleat Radio Rally

Thomas takes a look at another unlikely show supporting the Atari 8 bit range of computers

Gralin International decided to attend the 34th Radio Rally held at Longleat House, so I thought I would come along and help out at the User Group stand. First of all I didn't have a clue what a Radio Rally is, I figured it had something to do with radio controlled cars and aeroplanes, you know these remote controlled things. How wrong I was, on arrival it looked more like a fun fair with seven huge tents build in a circle and a marching band rehearsing 'Jingo" on the grass in the middle. Well, I parked my old banger and entered the first tent, hoping to find the others. Luckily I was given a booklet with all the diagrams and locations of all groups attending but first I had to have a look because I was curious to find out what attracts people for over 30 years to this location. Having ventured inside the first tent, things became clear, or at least somewhat clearer. It was all about the "Radio Society of Great Britain" and we are talking HAM Radio, Packet Radio, RTTY and the like. At last I knew why we were here because anything a radio can do, a computer can do as well, only better. (Ever listened to radio one on your XL? No, me neither.)

Anyway, it was just like a huge market, people selling and/or buying gadgets for your very own radio station. Big antennas, mobile and hand held transceivers, power meters, a radio users dream come true here. Fighting my way through the crowd I encountered "PAGE 6", the only English 8-bit magazine on the market. After the usual "hello" and "haven't seen you for a long time" it was far too busy for a chat but I noticed that "PAGE 6" was selling some software to run your radio on the XL. Moving on, in the same tent I met "XL/XE ALIVE", now trading under the name of "GLADDEN HOUSE", they were selling some software on tape very cheap and also some old Atari Users Mags.

Okay, it was now time to find "GRALIN INT." and by studying my booklet I found our stand to be in the opposite tent, just next to a craft fair tent. The craft fair tent was nice, I found a little self made something for my better half and I

Ever listened to radio 1 on your XL?
No, me neither

even won a cuddly toy at the lottery.

Finally I arrived at our stand where I met the others already selling software and our supposed-to-be-quarterly newsletter 8-16. Graham had set up an XL running some picture demos loading from a hard disk and mentioned that the software was selling well and that we also had some very good feedback from people showing a lot of interest. Our stand was quite big but it was crowded behind the counter with five people staggering about and not to forget the guys either side behind their stands walking all over the place, it was not the ideal situation, also being in a tent it gets somewhat stuffy after a while, never mind.

In the late afternoon Colin and I went for a little stroll, things had calmed down by then and it started to rain a little. PAGE 6, by that time, sold some software for as cheap as 0.50 pence and I bought a light gun game called "Gangsterville" and very good it is, too. He said the show went well and almost sold out, shame he didn't bring any PD software along as well.

Walking along I saw that it was not only radio related items the exhibitors had for sale, but also new or second hand TV's, VCR's, saw an old APPLE II for 30 pounds, half priced SEGA or NINTENDO carts, so it was really something for everybody, like a huge car boot sale held in tents. So, after a long and exciting day it had proved very successful for Gralin International and I guess for all the others as well. It was time to break up and I must apologise that I did not help to pack everything but I would have only been in the way.

Overall, it was a very well organised show in a posh location (Dig that house).

Thomas Holzer

PS. The band was pretty awful!

PPS. The WP used for writing this article was "Broderbund's" Bank Street Writer.

Up 'n' Coming Events

AMS5
9th November 1991
Bingley Hall, Stafford
Sharward Services, 0473 272002

Computer Shopper Show
5-8th December 1991
Wembley Conference Centre
081 868 4466

16 Bit Computer Show
7-9th Feburary 1992
Novotel Hotel, Hammersmith, London
081 549 3444

All Formats Fair's
3rd November 1991
Royal Horticultral Hall, London
10th November 1991
National Motorcycle Museum, B'ham
1st December 1991
City Hall, Glasgow
14th December 1991
Royal Horticultral Hall, London
15th December 1991
University of Leeds Sports Centre

Local Groups (see map below)

Name: **Atari User Group Of Ireland**
Contact: Mike Casey
3 St. Kevins Park, Kilmacud, Co Dublin
Notes: XL-ST-Others, Meetings, Newsletter, PD

Name: **Bloxwich Computer Club**
Contact: Edward Hunt 0922 409291
28 Station Street, Bloxwich, Walsall WS3 2PD
Notes: ST-Others, Meetings, PD

Name: **Bournemouth & Poole Atari User Group (BaPAUG)**
Contact: Ian Brooker; 163, Verity Crescent, Canford Heath, Poole, Dorset, BH17 7TX
Meetings: 1st Friday every month at the Kinson Community Centre, Pelhams, Millhams Lane Kinson
Newsletter: 8·16 (You're reading it)

Name: **Cheshunt Computer Club**
Contact: Derryck Croker 0923 673719
196 Coates Way, Garston, Watford, Herts WD2 6PE
Notes: ST-Others, Meetings

Name: **Mid-Cornwall Co-Op Computer Club**
Contact: Mike Richards 0726 890473
8 Victoria Road, Roche, St. Austell, Cornwell PL26 8JF
Notes: ST-Others, Meetings

Name: **Norwich User Group**
Contact: Ken Ward 0603 661149
45 Coleburn Road, Lakenham, Norwich NR1 2NZ
Meetings: 1st Sunday every month. Contact Ken for time & place.

Name: **South West ST User Group**
Contact: David Every
5 Turbill Gardens, Chaddlewood, Plympton, Plymouth, Devon, PL7 3XP
Notes: XL-ST, Meetings, Newsletter, BBS, PD

Name: **Swindon Computer Club**
Contact: Mike Bird 0793 539105
46 Eastcott Road, Swindon, Wilts SN1 3LR
Notes: XL-ST-Others, Meetings PD

Name: **The Friday Club**
Contact: Nicholas Bavington (0908) 612272
8 Byron Drive, Newport Pagnel, Bucks MK16 8DX
Meetings: Every Friday at Ousedale School Physics Dept OR a members house.
Notes: XL-ST, Hardware & Software development.

Name: **Wigan Computer Club**
Contact: Alan Owen 0942 212662
1 Lidgate Close, Wigan Lancs WN3 6HA
Notes: ST-Others, Meetings, Newsletter, PD

Name: **XL/XE Alive**
Contact: Bill Sutton 0784 255894
13 St. Annes Avenue, Stanwell, Middlesex TW19 7RL
Notes: 8-bit

Special Interest Groups

Name: **GFA User Magazine**
Address: 186 Holland Street, Crewe, Cheshire CW1 3SJ
Telephone: 0270 256429

National Groups

Name: **Association of Atari User Groups**
Address: 45 Coleburn Road, Lakenham, Norwich NR1 2NZ
Telephone: 0603 - 661149

International Groups

Name: **Club Cenacle**
Address: B.P 49, 95110 Sannois, France

Name: **Genesee Atari Group**
Address: PO Box E, Flint, MI 48507 USA

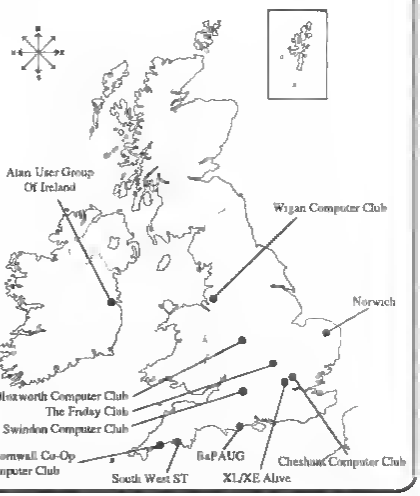
Name: **Johannesburg Atari Computer Enthusiasts (JACE)**
Address: 2 Whitehall Street, Hurst Hill Johannesburg, South Africa, 2092

Name: **Marylaod Atari Computer Club (MACC)**
Address: 8591 Wheatfield Way Ellicott City, Maryland 21043, USA
Telephone: (301) 461 7556
Newsletter: M.A.C.C. News

Name: **North East Atari Team (NEAT)**
Address: P.O. Box 18150 0150, Phila., Pa. 19116, USA
Newsletter: The Atarian

Name: **Northern Virginia's Atari Computer User Group**
Address: 8612 Thames St. Springfield, VA 22151 USA

Name: **Pittsburgh Atari Computer Enthusiasts (PACE)**
Address: P.O. Box 13435. Pittsburgh, PA 15243. USA



Notice To All User Groups

If you run or belong to a user group that supports any of the Atari range of products and wish your group to be listed please forward details to the BaPAUG and AAUG

GRALIN INTERNATIONAL

11 Shillito Road, Parkstone, Poole, Dorset

Re-Conditioned
800XLs & 65XEs
Keyboard units only - NO PSU
Limited Quantity
£22.50 each

See ya at AMS5!

Zeppelin Software On Disk

Zybex / Speed Ace / Las Vegas Casino / Phantom
Ninja Commando / Mirax Force / Jocky Wilson's Darts
Fantastic Soccer / Mountain Bike Racer / Fred
World Soccer / Bhnky's Scary School / Fruit Machine
Kenny Dalglish / Cavernia / Stack Up / Mission Shark
All titles only £4.99 each
Where possible please indicate alternative title

APX Cassette Collections

£4.00 per set

Set #1: Utilities #1
Text Formatter (Forma); Sound Editor, Decision Maker; T Text Display, Chameleon CRT Terminal Emulator

Set #2: Games #1
Mugwump, Babel, Block 'em, Phobos

Set #3: Educational Software
Type Attack, Counter; Memory Match, Morse Code Tutor

Set #4: Games #2
Querxon Solitaire; Pro Bowling; Codecracker

Set #5: Utilities #2
Extended WSFN, Banner Generator; Instedit, Ultimate Renumber Utility, Dsembler

Set #6: Games #3
Blackjack Casino, Dice Poker; Mankala, Lookahead

Set #7: Games #4
Avalanche, Attank!, Space Trek, Salmon Run

Set #8: Mixed Bag
Player Piano, Terry; Space Chase, Magic Melody Sox

Gralin's Product Line

DIGIDrum II	£5 99
MIDI Master	£TBA
Multi-Viewterm + Interface	£29 99
Percusion Master	£TBA
Replay Sound Sampler	£TBA

8bit Hard Drive Interface

Only £99.95

The K-P-I Hard Drive interface will allow you to connect a 20MB SCSI drive (not supplied) to your Atari XL/XE computer It comes complete with the MyDOS disk operating system and can also be used with SpartaDOS To complete the system you will need to purchase a PSU (63watts minimum) and cables XE users also require the XE Adaptor board (£15 00) For more information sheet please send SSAE

New Software from Germany

LDS 'C' Emulator (Cass to Disk backup)	£6 95
LDS Freezer XL/XE (dump memory to disk)	£6 95
6502 MC-Monitor (D)	£12 95
Action! Tooldisk	£9 95

ST Xformer Interface

Atari 8bit emulator for Atari ST complete with cable for connection of 8bit disk drive, thus allowing you to boot those protected disks

£19.95

NEW Games

Kross	£4 95(D)
Scaremonger V2 5	£7 95(D)
Invasion	£6 95(D)
Barkonid	£9 95(D) £6 95(C)
Player s Dream 1	£6 95(D)
Lighttraces	£6 95(D)
Mad Marbles	£4 95(D)
Glaggs It!	£6 95(D)
Taipei XL/XE	£6 95(D)
Rubber Ball	£9 95(D)

P&P / Handling Charges

Software £1 50 for orders under £50
Hardware £2 50 per order

Overseas orders extra please ask for quote Cheque / Postal Orders only.

Cassette Software £2 49 per set

Set 1: Star Raiders, Chess, Asteroids, Missile Command, Centipede, Airstrike 2, Realsport Tennis, Chop Suey, Realsport Football, Basketball

Set 2: Type Attack, Missile Command, Star Raiders, Centipede, Realsport Tennis, An Invitation To Programming

Lynx Games Cartridges

All titles are £28 49 each

A.P.B / Blockout / Blue Lightning / California Games / Checkered Flag Chips Challenge / Electro Cop / Gates of Zendecon / Gauntlet III Grid Runner / Klax / Ms Pacman / NFL Football / Ninja Gaiden Pacland / Paperboy / Rampage / Roadblaster / Robo-Squash Rolling Thunder / Rygar / Scrapyard Dog / Shanghai / Shime World Tournament Cyberball / Turbo Sub / Vindicators / War Birds World Class Soccer / Xenophobe / Xybots / Zalor Mercenary
Where possible please indicate alternative titles

Lynx Accessories

ComLynx Cable	£9 99	Lynx Kit Case	£14 99
Lynx Sun Screen	£5 99	Lynx Pouch	£9 99